

81

100-443887-100

(10) International Publication Number
WO 01/31450 A1

[Continued on next page]

(57) Abstract: An analysis scheduler for scheduling the automatic processing of performance data through a plurality of analysis tools is disclosed. Performance data provided to some of the tools by the analysis scheduler may be specified to be within a predetermined (but variable) look-back period. The analysis tools identify faults and anomalous conditions and also create repair recommendations, and automatically create problem cases when conditions warrant, or update existing problem cases with additional data, all under control of the analysis scheduler. The problem cases are reviewed by a human user and then forwarded to the railroad for implementation. A process for determining which faults are regarded as critical is also disclosed.

BEST AVAILABLE COPY



patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

With international search report.

APPARATUS AND METHOD FOR PERFORMANCE AND FAULT DATA ANALYSIS

5

BACKGROUND OF THE INVENTION

This invention relates to a method and apparatus for automatically analyzing parametric performance and fault related data from a vehicle or a machine, such as a railroad locomotive or a fleet of railroad locomotives.

10 A locomotive is one example of a complex electromechanical system comprised of several complex subsystems. Each of these subsystems is built from components which over time will fail. When a component does fail, it is frequently difficult to determine the cause of the failed component because the effects or problems that the failure has on the subsystem are often neither readily apparent in
15 terms of their source nor are they typically unique. The ability to automatically diagnose problems that have occurred or will occur in the locomotive subsystems has a positive impact on minimizing locomotive downtime. Downtime of any machine or vehicle can be advantageously impacted by accurate and early diagnosis.

It is also known that cost efficient operation of a railroad or any vehicular
20 fleet, requires minimization of line-of-road failures and locomotive down time. Failure of a major vehicle subsystem can cause serious damage, costly repairs, and significant operational delays. A locomotive break-down while in service is an especially costly event, requiring the dispatch of a replacement locomotive to pull the train consist and possibly rendering a track segment out of service until the train is
25 moved. As a result, the health of the locomotive engine and its constituent subsystems is of significant concern.

Previous attempts to diagnose problems once they have occurred on a vehicle or other complex machine usually involve performing inspections by experienced personnel who have in-depth individual training and experience with the vehicle
30 locomotives. Typically, these experienced individuals use available information that has been recorded in a log. Looking through the log, they use their accumulated experience and training in mapping incidents occurring in systems to problems that

may be causing the incidents. If the incident-problem scenario is simple, then this approach works fairly well. However, if the incident-problem scenario is complex, then it is very difficult to diagnose and correct any failures associated with the incidents.

5 Currently, computer-based systems are being used to automatically diagnose problems in a vehicle to overcome some of the disadvantages associated with relying completely on experienced personnel. Typically, a computer-based system utilizes a mapping between the observed failure symptoms and the equipment problems, using techniques such as table look ups, a symptom-problem matrices, and production rules.

10 These techniques work well for simplified systems having simple mappings between symptoms and problems. However, complex equipment and process diagnostics seldom have such simple correspondences. In addition, not all symptoms are necessarily present if a problem has occurred, thus making other approaches more cumbersome.

15 The above-mentioned approaches either take a considerable amount of time before failures are diagnosed, or provide less than reliable results, or are unable to work well in complex systems. There is a need to be able to quickly and efficiently determine the cause of any failures occurring in the vehicle systems, while minimizing the need for human intervention.

20 There is also no automatic or systematic mechanism for the identification of incipient machine problems. Instead, conventionally, the owners have relied on regular inspections and the observation of performance anomalies by the operator. Some cursory inspection processes may be accomplished while the machine or vehicle is in service; more thorough inspections require it to be taken out of service
25 for several days. In any case, machine or vehicle down time, whether for inspection or repair, represents a significant operational cost. The avoidance of these costs by accurate fault diagnosis and prediction of potential failures represents an important cost saving opportunity for the machine owners and operators.

30 As a further means to reduce downtime, there has been a focus on the engineering design process with an objective of increasing the mean time between failures for subsystems and components. While this is certainly a commendable objective, it remains for the owners to continue their cost containment goals through

the collection and monitoring of real time performance data and fault related information directly from the vehicle, and the implementation of repairs before the problem requires significant down time.

5 BRIEF SUMMARY OF THE INVENTION

Generally speaking, the present invention fulfills the foregoing needs by providing a method for analyzing real-time performance data and for identifying a plurality of faults and critical faults in machines. The method generally includes collecting from a predetermined plurality of the machines respective machine data
10 indicative of performance over a predetermined period of time. The method further includes respective identifying steps that allow for identifying in the collected machine data respective faults most frequently occurring relative to one another and for identifying in the most frequently occurring faults, respective faults that, relative to one another, affect a higher number of machines. A classifying step allows for
15 classifying the faults identified in the last-recited identifying step based on an expected level of machine degradation associated with the identified faults. A storing step allows for storing in a database of critical faults, any faults classified as likely to result in an imminent machine mission failure.

The system further includes means for identifying in the collected machine
20 data respective faults most frequently occurring relative to one another. There is also provided means for identifying in the most frequently occurring faults, respective faults that, relative to one another, affect a higher number of machines. Classifying means allows for classifying the faults identified with the last-recited identifying means based on an expected level of machine degradation associated with the
25 identified faults. A database is coupled to the means for classifying to store any faults classified as likely to result in an imminent machine mission failure, the stored faults comprising the plurality of critical faults.

In one application of the present invention, each locomotive in a railroad's fleet of locomotives includes an on-board monitor for collecting real-time operational
30 data. The on-board monitor transmits performance and fault data on a regular basis to a monitoring and diagnostic service center, where the present invention analyzes the received data. There could be as many as 3,000 locomotives in a fleet, each reporting

data on a daily basis. Such an enormous amount of data will easily overload a human operator. It is thus necessary to automate the execution of analysis tools so that the analysis of fault and parametric performance data from the automated data downloads can be accomplished in an efficient and productive manner.

5 In accordance with the teachings of the present invention, the scheduling and execution of each fault analysis tool occurs without human intervention and is based upon dynamic and time-critical criteria applied to the received data. For instance, one such criterion could be the priority of the queued data awaiting analysis. The present invention automatically schedules, prioritizes, and oversees the execution of one or
10 more analysis and diagnostic tools for analyzing the locomotive data. The analysis scheduler of the present invention also conducts on-line monitoring of the downloaded data, prioritizes the data queued for each analysis tool, and ensures that all prerequisites are met before triggering execution of an analysis or diagnostic tool. In one embodiment, for example, there may be limits on the number of instances of
15 each tool that can be executed simultaneously, and the limits may be dependent on the priority of the data. For instance, one limit applies to normal priority data and a second limit applies to high priority data. The analysis scheduler maintains and enforces these limits. In the event that a system outage occurs, the analysis scheduler automatically restarts each analysis tool that was processing when the outage
20 occurred. In the event of an analysis tool failure, the automatic scheduler retries operation of the failed tool, until a predetermined retry limit is reached.

Following execution of the analysis tools, the present invention creates a problem case for each problem or anomaly identified by the automated analysis process. To focus the limited human resources on actual problem solving, it is
25 desirable to automatically create the problem cases. The problem case incorporates the outputs from the multiple analysis and diagnostic tools and includes all relevant data, including symptoms, the nature of any fault, related performance parameters, diagnostic information, and repair recommendations as generated by the automated analysis process. The problem case generator displays all this information visually
30 for viewing by a human diagnosis and repair expert.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention can be more easily understood and the further advantages and uses thereof more readily apparent, when considered in view of the description of the preferred embodiments and the following figures in which:

5 Figure 1 shows an exemplary machine, e.g., a locomotive, that may readily benefit from the teachings of the present invention;

 Figure 2 illustrates the peripheral devices with which the analysis scheduler of the present invention communicates;

 Figure 3 depicts a microprocessor implementation of the present invention;

10 Figures 4, 5A, 5B, 6, and 7 illustrate subprocesses of the analysis scheduler of the present invention;

 Figure 8 illustrates the case creation process of the present invention;

 Figures 9A and 9B are software flow charts depicting the case creation process;

15 Figures 10, 11, 12, 13, 14 and 15 illustrate the operation of the analysis and diagnostic tools shown in Figure 8;

 Figure 16 illustrates the case repetition detection process of the present invention; and

 Figure 17 is a flow chart illustrating the identification of critical faults.

20

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

 Figure 1 shows a schematic of an exemplary locomotive 2. The locomotive may be either an AC or DC locomotive. The locomotive 2 is comprised of several complex systems, each performing separate functions. Some of the systems and their
25 functions are listed below. Note that the locomotive 2 is comprised of many other systems and that the present invention is not limited to the systems disclosed herein.

 An air and air brake system 4 provides compressed air to the locomotive, which uses the compressed air to actuate the air brakes on the locomotive and cars behind it.

30 An auxiliary alternator system 5 powers all auxiliary equipment. In particular, it supplies power directly to an auxiliary blower motor and an exhaustor motor. Other equipment in the locomotive is powered through a cycle skipper.

A battery supplies power to a cranks system 6 to start operation of a diesel engine for operation of a DC bus and a HVAC system. The DC bus in turn provides voltage to maintain the battery at an optimum charge.

5 An intra-consist communications system collects, distributes, and displays consist data across all locomotives in the consist.

A cab signal system 7 links the wayside to the train control system. In particular, the system 7 receives coded signals from the rails through track receivers located on the front and rear of the locomotive. The information received is used to inform the locomotive operator of the speed limit and operating mode.

10 A distributed power control system provides remote control capability of multiple locomotive consists anywhere in the train. It also provides for control of tractive power in motoring and braking, as well as air brake control.

15 An engine cooling system 8 provides the means by which the engine and other components reject heat to the cooling water. In addition, it minimizes engine thermal cycling by maintaining an optimal engine temperature throughout the load range and prevents overheating in tunnels.

An end of train system provides communication between the locomotive cab and last car, via a radio link, for the purpose of emergency braking.

20 An equipment ventilation system 9 provides the means to cool the locomotive equipment.

An event recorder system records FRA (Federal Railroad Administration) required data and limited defined data for operator evaluation and accident investigation. It can store up to 72 hours of data, for example.

25 A fuel monitoring system provides means for monitoring the fuel level and relaying the information to the crew.

An exemplary global positioning system uses satellite signals to provide accurate position, velocity and altitude measurements to the various control systems. In addition, it also provides a precise UTC reference to the control system.

30 A mobile communications package system provides the main data link between the locomotive and the wayside via a suitable radio, (e.g., a 900 MHz radio).

A propulsion system 10 provides the means to move the locomotive. It also includes the traction motors and dynamic braking capability. In particular, the

propulsion system 10 receives power from the traction alternator and through the traction motors converts it to locomotive movement.

A shared resources system includes the I/O communication devices, which are shared by multiple systems.

5 A traction alternator system 11 converts mechanical power to electrical power which is then provided to the propulsion system.

A vehicle control system reads operator inputs and determines the locomotive operating modes.

10 The above-mentioned systems are monitored by an on-board monitor (OBM) system 12. The OBM system 12 tracks incidents occurring in the systems with an incident log. Locomotive 2 may optionally include an on-board diagnostic system 13, such as described in greater detail in U.S. Patent No. 5,845,272.

15 Figure 2 shows an analysis scheduler 15 and the various tables and data bases with which it communicates. The analysis scheduler 15 is implemented as a computing device as illustrated in Figure 3. The elements of the computing device are well known to those skilled in the art and include a microprocessor 16, a non-volatile memory 17, a RAM 18, and an input/output interface 19. The structure and operation of these devices are conventional in all respects and well known.

20 Returning to Figure 2, a download module 20 receives performance and fault data, for instance from an on-board monitor system 12. The download module 20 receives the performance and fault data, creates a download case that includes that downloaded data, and inputs the download case to a performance data table 21 and a fault data table 22. Data downloaded during a download session with a given locomotive is automatically assembled into a download case in accordance with the
25 teachings of the present invention. Information relevant to the download case can later be "attached" to this download case and its subcases. A different download case is created each time that data is downloaded from a locomotive.

30 The download module 20 also adds a record to a download status table 24 when the loading of fault or other data to the performance and fault data table 22 is complete. The analysis scheduler 15 monitors the download status table 24 and activates the various analysis and diagnostic tools, as will be discussed further herein below, when the data needed by those tools is available in the performance and fault

data table 22, or in other words, when creation of the download case is complete. The analysis scheduler 15 deletes entries in the download status table 24 when tool execution on the downloaded data has been scheduled, i.e., when a record has been created in a queue 34. In one embodiment, each tool has a unique queue, although
5 that is not specifically shown in Figure 1.

Specifically, when the download status table 24 indicates that a particular download of data is complete, the analysis scheduler 15 creates a download tool subcase, i.e., a subcase to the download case, for each tool that will process the downloaded data and then dispatches a record of that download tool subcase to the
10 queue 34 for that specific tool. The subcases are stored in a subcase table 26. The actual tool execution on specific data is prioritized by the type of data in the queue 34 and also the time when the download tool subcase record was created. Certain types of downloaded performance or fault data will have a higher priority than others. The analysis scheduler 10 spawns the tools, as will be discussed below. As the various
15 analysis tools process the downloaded data, they create output data of their own. Each download tool subcase is augmented with this output data from the associated analysis tool, as well as status information about the progress of the tool's execution.

The analysis scheduler 15 updates tool execution information in the download tool subcases, as stored in the subcase table 26. Each download tool subcase includes
20 a tool status entry indicating the execution status of each tool. For instance, a single tool can be running simultaneously on four different packets of performance and fault data, i.e., download cases. Each of these four executions will likely be at a different point in the tool execution process, and tool execution can take up to several minutes, dependent upon the amount of data to be processed and the specific tool. Thus, the
25 download tool subcase reflects the running status of each tool for each simultaneous instantiation for the tool. Included among these status indicators are: execution not started, tool has exceeded its retry limit, tool has exceeded its execution time limit, tool execution completed normally, and a series of sequential values, wherein each value in the sequence indicates the current point on the tool execution timeline. The
30 analysis scheduler 15, by checking the download tool subcase in the subcase table 26, can detect when a specific tool execution is complete, has failed, or has terminated before completion.

A tool execution table 28 contains a record for each tool, including tool operational parameters, such as execution limits and execution prerequisites. One of these parameters sets a limit on the number of simultaneous instantiations for the tool when a normal-priority execution is next in the queue. There is also a separate instantiation limit applicable to high priority tasks in the queue. The tool execution table 28 also includes various prerequisite value requirements, for example, a requirement that a certain tool must be run before another tool can process the data. Queues are monitored and tools activated in accordance with these controls stored in the tool execution table 28. When the number of executing tasks falls below the normal priority limit, the next task (in priority order) in the queue will be spawned. If a high priority task is in the queue, then the normal priority limit is ignored in favor of the high priority task. So long as the high priority limit is not exceeded, the high priority task is activated for processing.

A configuration table 30 stores information indicating which tools (and which versions thereof) should be run for download cases from a specific locomotive road number. For example, tool version number changes represent changes in the tool software, which may render the tool incompatible with certain downloaded data. As a result, the downloaded data may first require modification units conversion, for example, prior to running the tool on the data. The configuration table 30 also includes the file location of the tool executables.

Each download case is stored in the download case table 32. As discussed above, each download case includes the specific performance and fault data from a locomotive. In the download tool subcases, the downloaded information is bundled into a package for execution by one of the diagnosis or analysis tools. The individual download tool subcases created under each download case include the performance and fault data to be processed by the tool. After a download tool subcase has been created and appropriate entries made in the subcase table 26, the analysis scheduler 15 moves the download tool subcase to a queue 34. From here, the download tool subcase will be executed by the identified tool, when processing reaches that point in the queue 34. Figure 2 also illustrates that the analysis scheduler 15 controls the running of the tools, after all the pertinent information is available. This is shown generally by the box bearing reference character 36 and will be discussed in greater

detail in conjunction with the flow charts of Figures 4 – 7. Also, operation of a problem case generator 31 will be discussed herein below. Once a download tool subcase is completed (i.e., processing of the data by the tool is finished) then the analysis scheduler 10 closes that download tool subcase in the download case table

5 32.

Figure 4 illustrates the process executed by the analysis scheduler 15 in preparation for running a diagnostic or analysis tool. Processing begins at a start step 40 and continues to a decision step 42 where inquiry is made as to whether one or more records in the download status table 24 indicate that processing has not yet been initiated on a download case, i.e., performance or fault data received from the locomotive. If the result of decision step 42 is true, processing moves to a step 44 where the entry corresponding to the highest priority data is selected for processing. At a step 46, the analysis scheduler 15 locates the associated download case in the download case table 32, the necessary tool configuration and execution information from the configuration table 30 and the tool execution table 28. At a step 48, the analysis scheduler 15 creates the download tool subcase records in the subcase table 26 (based on information in the configuration table 30 and the tool execution table 28) and moves the pertinent information to the queue 34. Now that the information has been queued, at a step 50 the analysis scheduler 15 deletes the associated download case record in the download status table 24. A commit step 51 ensures that the modifications made at the steps 48 and 50 update the appropriate tables simultaneously. Processing then returns to the decision step 42, for retrieval of additional download cases.

If the result from the decision step 42 is false, at a step 52 the analysis scheduler 15 retrieves the sleep time from a system parameter table 23 of Figure 2 and then falls into a sleep mode, as indicated at a step 53. When the sleep time has expired, processing returns to the decision step 42, where records in the download status table 42 are again checked.

Figures 5A and 5B are the process flow charts for the tool spawning software, which launches the execution of each of the diagnostic and analysis tools. The tool spawning software is another software component of and is executed by the analysis scheduler 15. There is one tool spawner for each tool, as executed by the analysis

scheduler 15. Processing begins at a start step 58 where the specific tool identification is input to the tool spawner process. At a step 59, the spawner sleep time is retrieved and at a step 60 the tool execution parameters (from the tool execution table 28 and the configuration table 30) are input to the tool spawner. For instance, one of these parameters is the number of permitted high-priority simultaneous tool executions. Processing then moves to a step 61 where the highest priority download tool subcase for which a retry is required (i.e., where the retry flag value is one) is selected. A retry would be required, for instance, if the system crashed while the tool was processing download tool subcase data. The setting of the retry flag will be discussed further in conjunction with Figure 5. Processing then moves to a decision step 62 where the selection count is checked. If data was selected at the step 61, then the selection count will have a non-zero value and processing continues to a step 63 where the tool execution is spawned (i.e., the tool processes the data).

15 If no data was selected at the step 61, because there are no download tool subcase records awaiting a retry execution, the result of the decision step 62 is true and processing moves to a step 64. At the step 64, the tool spawner counts the number of download tool subcases that are currently running under that tool, and sets an execution count value equal to that number. Processing then moves to a step 65 where the tool spawner selects the high priority download tool subcases in the queue 20 34 for which all prerequisites have been met (including the completion of the download process) but having a retry flag value of zero. At a decision step 66, the selection count from the step 65 is checked. If a download tool subcase record was selected at the step 65, the result of step 66 will be false and processing moves to a decision step 67. Here the execution count (representing the number of in-process download tool cases) is compared to the simultaneous limit for high priority cases. If the latter value is greater than or equal to the former, then no additional tool executions can be spawned and processing moves to the sleep step 68. Note the sleep time for each specific tool was previously input to the tool spawner at the step 59.

30 After the sleep time has elapsed, processing returns to the step 64. If the simultaneous execution limit is not exceeded, the decision step 67 produces a true result and the tool is spawned at a step 73.

If no "high priority" download tool subcases were selected at the step 65, the result of the decision step 66 is true and processing moves to a step 69. Here the normal priority download tool subcases are examined to determine whether any have all prerequisites satisfied and are therefore ready to be run. At a decision step 70, the selection value set at the step 69 is checked and if it is zero, indicating that there were no selections made at the step 69, processing moves to the sleep step 71. After the sleep time has elapsed, processing moves from the sleep step 71 back to the step 64.

If the step 69 resulted in a selection, the result from the decision step 70 is false and processing moves to a decision step 72. Here the executing count is compared to the simultaneous execution limit for normal priority download tool subcases. If the result is true, processing moves to the step 73 where the tool run is spawned. If the result of the decision step 72 is false, the process sleeps, as indicated by a sleep step 74. Upon reawakening, processing returns to the step 64. Following the step 73, processing returns to the step 64 to again set the execution count and select cases for tool spawning at the steps 65 and 69.

The tool monitor software process, illustrated in Figure 6, monitors the execution of the various tools associated with the present invention to ensure that tool execution is not exceeding any tool limits. Processing begins at a start step 77 and continues to a step 78 where the monitor sleep time is obtained from the configuration table 30 and a value is set to indicate the initial pass through the tool monitor. At a step 79, tool execution parameters are obtained from the tool execution table 28. At a step 80, the download tool subcases are selected in priority order among those download tool subcases that are executing. From the step 80, processing moves to a step 81 where the next (or first, as the case may be) download tool subcase in the list is obtained. At a decision step 82, a check is made to determine whether the download tool subcase is processing. If the download tool subcase is executing, processing moves to a decision step 83. If this is the first time for processing the download tool subcase, the decision from the decision step 83 is true. Processing then continues to a step 84 where the retry flag is set to one and the download tool subcase is committed for execution by the tool. If this is not the first processing attempt, the result of the decision step 83 is false and at a decision step 85, the restart count (which is a value contained within the download tool subcase record) is compared to the retry

limit. If the count is greater than or equal to the retry limit, then the execution is terminated at a step 86 and the subcase is removed from the queue. If the retry limit has not been reached, processing moves from the decision step 85 back to the step 84 and the subcase is queued for processing through the tool.

5 If the download tool subcase is not currently processing, the result from the decision step 82 is false. Processing then moves to a step 87 where the processing lapsed time is calculated. At a decision step 88, the processing time is compared to the processing time limit. If the limit has been exceeded, processing moves to the step 86 where execution is terminated. Whenever an execution is terminated, a record
10 is created calling this occurrence to the attention of the system user. At this point, human intervention is required to resolve the processing problem. For example, processing may have been unsuccessful due to corrupted data in the download case. If the result from the decision step 88 is false, the tool continues to execute the subcase and the tool monitor process moves to a decision step 89. Note that
15 processing also moves to the decision step 89, after the step 84. At the decision step 89, if the end of the list has not been reached, processing returns to the step 81 for fetching the next download tool subcase. If the end of list has been reached, the first time flag is set to "no" at a step 90. The tool monitor then sleeps, as indicated at a step 91. When the sleep time expires, processing returns to the step 80 where the
20 download tool subcases are again retrieved.

 The analysis scheduler 15 also closes tool execution after processing all the download tool subcases. Figure 7 illustrates the software steps for a download case closer program executed by the analysis scheduler 15. Processing begins at a start step 94 and continues to a decision step 95. Here the download cases are examined to
25 determine whether any indicate that both the fault and parameter downloads have been completed and all the download tool subcases under the download case have been closed following tool processing of the download tool subcase, or the download has failed due to a communications problem. If either of these statements is true, processing moves to a step 96 where the download case is closed. Once all download
30 tool subcases have been closed, the corresponding download case can be closed. Also, if the download of data from the locomotive has failed, the download case can be closed. If the response from the decision step 95 is false, the case closer

downloads the sleep time from the database at a step 97. The case closer program then sleeps, as indicated at a sleep step 98. At the end of the sleep time, processing returns to the decision step 95. In one embodiment of the present invention, the sleep time is 24 hours.

5 Figure 8 illustrates the process utilized by the present invention for creating an analysis and repair case, otherwise referred to as a problem case. A problem case is a collection of information relevant to one or more performance anomalies or faults. For instance, as applied to the present invention, the case includes output information from the various analysis and diagnostic tools, fault repair codes, identification of
10 critical faults, and anomaly code data associated with the downloaded case. By contrast to the problem case, the download case comprises the performance parametric information downloaded from the locomotive, or in other embodiments, the machine or vehicle. The problem case also includes repair and maintenance recommendations, again as determined by the analysis and diagnostic tools. All the
15 case information is available to a user, who is someone knowledgeable in the area of locomotive operation, faults and repairs. The user reviews the case to determine the accuracy of the information presented therein and may further append additional information. For instance, the user can add repair recommendations based on his experiences. Once the problem case is completed, the user transmits the case to
20 railroad maintenance and service personnel. This can be accomplished by simply calling the railroad or sending the case via electronic mail. The objective is to provide the problem case to the railroad so that the repair or maintenance recommendations included therein can be implemented in a timely fashion to avoid a locomotive break down, or to return an out-of-service locomotive to the fleet.

25 Most of the time the processing of a given download case is uneventful, and there are no noteworthy anomalies or faults. In such situations, no problem case is required, and the download case and download tool subcases are saved to journalize this. However, if one or more of the analysis tools detects an anomalous condition or fault, then a problem case is created, which serves to collect and summarize all of the
30 outputs of all the analysis tools.

Figure 8 shows a locomotive providing three different types of information to a problem case creation system constructed according to the teachings of the present

invention. As discussed in greater detail in the patent applicable entitled "On-board Monitor for a Railroad Locomotive", cited above, certain severe faults within the locomotive immediately generate a call home (as designated by reference character 101) to the monitoring and diagnostic service center, where the problem case creation system resides. These faults are either severe in nature or require immediate attention and thus create a problem case directly, as indicated by a create case step 102. (See also the case generator 37 in Figure 1). To create the problem case, the call home process initiates a fault data download and a monitored parameter download as shown at a step 104. The problem case is then created at the step 102. Later, after the fault and monitored parameter information has been analyzed by the diagnostic tools, the results thereof will likely be added to the problem case created by the call home sequence. It is possible, however, that a new problem case, derived solely from the downloaded data, may also be created.

As discussed above in conjunction with the analysis scheduler 15, a step 106 depicts the downloading of fault data from the locomotive to the monitoring and diagnostic center where the analysis process occurs. In one embodiment, fault data is downloaded at least daily. It is possible, however, that there may be no fault data to download and in this case the fault tools are not run as there is no input data to be analyzed. Once the fault data is downloaded, processing moves to a step 108 where the scheduler analysis process is executed as discussed above in conjunction with the analysis scheduler 10. At steps 110, 112, 114, and 116, the case-based reasoning (CBR), Bayesian belief network (BBN), fault classification, and data pack anomaly detection (DPAD) tools are run, respectively. These tools are examples of fault and data analysis tools that can be utilized in conjunction with the present invention. Those skilled in the art recognize that other similar analysis tools are available. These tools, which were referred to generally by the process step 36 in Figure 2, will be discussed in further detail below. Although not shown in Figure 8, there is a data queue associated with each of the tools depicted. These queues hold the data until the tool is available for execution. Essentially, each tool analyzes the data based on different rules and metrics, including historical cases (faults, repairs and operational parametric information) and artificial intelligence schemes, to determine the nature of the fault and identify specific repairs (by repair code) that can be implemented to

alleviate the fault. The tools may also identify incipient problems within the locomotive, and thus allow the railroad to take corrective action before the problem becomes more severe.

Although the tools are shown executing in a parallel fashion in Figure 7, as is known to those skilled in the art, this is not a mandatory requirement. Other embodiments of the present invention include execution alternatives. For instance, the tools can run serially or in parallel after the downloaded case has been created (i.e., all the pertinent data is available for tool execution) and each of the download tool subcases have been created, or each tool can run independently after the download tool subcase for that specific tool is available. After all tools have processed the data, the case repetition detection step as illustrated by reference character 118 is executed. Finally, each tool can execute independently after its download tool subcase is completed and then immediately execute the case repetition detection step 118. The selection of one of these alternatives is not crucial to the essential scope or function of the present invention.

The tool spawner component (illustrated in Figures 5A and 5B) of the present invention controls the execution sequence of the tools illustrated in Figure 8. Of course, a tool cannot execute until the prerequisite information has been collected, i.e., until the download tool subcase is complete. The tool execution table 28 illustrated in Figure 2 stores the conditions that must be met before a specific tool can be run. The case repetition detection tool (see reference character 118 of Figure 8) is an additional tool of the present invention for which information is included in the tool execution table 28. The case repetition detection tool 118 is run to detect repetitive cases after one or more of the other analysis tools has executed. The case repetition detection tools will be discussed further herein below in conjunction with Figures 9A and 9B.

Whenever a new problem case is created, as indicated by the step 102 of Figure 8, certain information is entered into case fields to assist the expert user at the monitoring and diagnostic service center in analyzing the problem case. The information in the case fields may include: fault codes and descriptions of systems they indicate, repair codes and descriptions of repairs indicated, anomaly codes and descriptions of the warnings indicated, monitored parametric values associated with

faults, repairs and anomalies, probability or weighting factors associated with the indicated codes (where the weighting factors indicate the probability that the indicated repair will solve the indicated fault), the date, time during which the data was collected, and locomotive road number from which the data was collected.

5 Returning to Figure 8, in addition to fault data, download portrayed by the step 106, parametric performance data is also downloaded from the locomotive, as identified at a step 124. Analysis scheduler processing occurs, (see a step 126), as discussed in conjunction with Figures 2-7, when the download is complete. The step 126 is followed by running of the anomaly detection tool, (illustrated by a step 128),
10 and running of a trend tool (illustrated by a step 130). The case repetition detection program is then run at the step 118. If necessary, a case is created at the step 102 as discussed herein above.

 The flow charts of Figure 9A and 9B illustrate the algorithm for creating a case in accordance with the present invention, combining the features of the step 118
15 (run case repetition detection) and 102 (create problem case). Processing begins at a step 160, depicting the data download process. From the step 160, processing moves to both a step 162 and a step 164. At the step 162, the case-based reasoning (CBR), Bayesian belief network (BBN), and data pack anomaly (DPAD) tools are executed for the purpose of developing a problem case and advantageously for developing
20 repair recommendations for that problem case. The execution of these tools will be discussed in detail below. At the step 162, the tools are run using their normal look-back time period. As will be discussed further herein below, the look-back time is that period measured from the present to a point in the past, during which data collected will be processed by the tool. For instance, in one example, the look-back
25 period is seven days. Therefore, the tool will analyze fault data provided during the past seven days in an attempt to classify faults and develop repair recommendations. From the step 162, processing moves to a decision step 166 for determining whether any repair recommendations have been generated by the case based reasoning, the Bayesian belief network, or the data pack anomaly detection tool. If no such repairs
30 have been recommended, then this stage of the processing is completed, as illustrated by a step 168. If repairs were recommended, processing moves from the decision step 166 to another decision step 170, where the process determines whether there are any

existing closed or recommended problem cases. Closed cases are those for which the repair recommendations have been implemented by the railroad. Recommended cases are those where the repair recommendations have been transmitted to the railroad, and thus, in a sense, are no longer subject to changes or additions by expert personnel at the monitoring and diagnostic service center. Only open cases can be augmented by information from the current execution of the analysis and diagnostic tools. If there are no closed or recommended cases, processing moves to a step 172 where the repairs recommended by the tool are added to a repair list, identified in Figure 9A and 9B by a reference character 174.

10 If there are existing closed or recommended cases, then processing moves from the decision step 170 to a decision step 180. The decision step 180 determines whether any of the recommended repairs are identical to repairs in closed or recommended problem cases that were created within the look-back time frame. If there are no such identical repairs, then processing returns to the step 172 where these repairs are added to the repair list, where they may later be used to create a problem case. If all of the repairs are identical to repairs in closed or recommended problem cases, then it is necessary to change the look-back time so that only data collected after the most recently recommended or closed case is included in the tool analysis. In this way, the process ensures that only parameter and fault data collected after the most recent repair recommendation can generate a new problem case, because the data relied upon to create a previous closed or recommended problem case is no longer relevant for creating new problem cases. If the railroad has not yet performed a recommended repair, then the same kind of faults will be seen during the next download of fault and performance information resulting in generation of the same repair recommendations. The case repetition detection process (see reference character 118 of Figure 8) will then combine the current recommended problem case with existing recommended problem cases. This look-back time interval change is depicted by a step 182, where the look-back period is changed to begin immediately after the most recent recommended or closed case. At a step 184, the case based reasoning, Bayesian belief network, and data pack anomaly tools are re-run with the modified look-back parameter.

At a decision step 186, the process determines whether any repairs were recommended by the tool execution at the step 184, i.e., based on the tool re-run using the new look-back period. If no repairs were recommended, then this stage of the processing is completed, as depicted by a step 190. If there are any recommended repairs, they must be added to the repair list, as illustrated at a step 188.

Returning to the download data step 160, at a step 164 the anomaly detection (AD) and trend anomaly detection tools are run. Also, at a step 196 the fault classification and anomaly detection tools are executed. All anomalies found are added to an anomaly list 200 at a step 198.

From the step 164, after the trend anomaly tool is executed, processing moves to a decision step 202 to determine whether any anomalies were recommended. If none were recommended, processing terminates at a step 204. If anomalies were found and recommended, processing moves to a decision step 206 where, as before, the process determines whether any existing or recommended problem cases are open. If no such problem cases are open, processing moves to a step 208 where the new anomalies are added to the anomaly list 200. If there are existing closed or recommended problem cases, then from the step 206 processing continues to a decision step 210. Here a determination is made whether the trend anomalies detected are identical to any trend anomalies in closed or recommended problem cases. If there are no such identities, processing again moves to the step 208, where the trend anomalies are added to the anomaly list. If one or more of the anomalies are identical to anomalies listed in closed or recommended problem cases, processing moves to a step 212 where the anomaly trend tool is run again without use of the state file, which stores historic operational trends. This process of rerunning the tools without the state files removes the effect of anomalies that should have been addressed by prior recommended or closed cases. After the anomaly tool is re-run, at a decision step 214 a determination is made whether any anomalies were detected. If none were detected, processing ends at a step 216. If anomalies were detected, they are added to the anomalies list by processing through a step 208.

After repairs are added to the repair list 174 and anomalies are added to the anomaly list (represented by a reference character 200), processing moves to a decision step 222. Here, the process determines whether there are any open problem

cases. If there are no open problem cases at that point, a new case is created at a step 224 and processing terminates at a step 226. The new problem case contains all the anomalies from the anomaly list 200 and all repairs from the repair list 174.

Alternatively, if there are open problem cases, it must be determined whether
5 the repairs or anomalies can be added to them at a decision step 230. Here it is determined whether there are any open problem cases less than x hours old, where x is a threshold value assigned by the user. If such an open problem case is available, processing moves to a step 232 where all of the anomalies and repairs are added to the repair list for that problem case. Also, the download case from which the faults
10 and/or anomalies were derived is linked as a child to the open problem case. The same locomotive symptoms may appear in multiple downloads over many days and all such downloads should be linked to the same open problem case.

If there are no open cases less than x hours old, processing moves from the decision step 230 to a decision step 234 for determining whether there are any repairs
15 in the repair list 174. If there are none, then processing continues to the decision step 236 where it is determined whether all the anomalies are found in an open case. If the answer is no, processing moves to a step 238 where a new case containing all the anomalies is created. Processing then terminates at the step 226. If all the anomalies are already found in an open case, processing moves from the decision step 236 to a
20 step 242 where the download case from which the current anomalies were derived is linked as a child of that open problem case.

Returning to the decision step 234, if there are repairs in the repair list 174, processing moves to a decision step 244. Here, it is determined whether all of the repairs are identical to those in an open problem case. If that is a true statement,
25 processing returns to the step 242 where the download case is linked as a child to that open problem case. If all the repairs are not identical to those in an open problem case, processing moves from the decision step 244 to the step 224 where a new problem case is created. Processing then terminates at the step 226.

Figure 10 illustrates the operational characteristics of the case-based reasoning
30 tool as identified by reference character 299. In the context of the case-based reasoning tool, a "case" is a collection of faults, anomalies, recommended repairs, and operational parametric information aggregated for the purpose of comparing with

other "cases" to determine a recommended repair to resolve the fault. As discussed above, on the first pass, the case-based reasoning tool uses a standard look-back period of seven days. This can be modified for subsequent executions, also as discussed above, dependent upon whether there are any repairs identical to those recommended by the case-based reasoning tool in a closed or a recommended case. The case-based reasoning tool analyzes the fault data and combinations thereof, using information from the case-based reasoning case base 300.

The configuration table 30 (see Figure 2) identifies the version of the case-based reasoning tool 299 that is to run, based upon the locomotive road number from which the fault and parametric operational data was taken. Reference character 304 illustrates the fault and related operational parametric information input to the case-based reasoning tool 299. The fault data covers only the current look-back period and is noise reduced. Noise reduction is the process of eliminating known faults in the locomotive. For instance, when the locomotive is in idle state, certain measured parameters may be beyond a pre-established threshold and, therefore, falsely indicate the occurrence of a fault.

The configuration table 30 also provides the probability threshold used by the case-based reasoning tool as a probability limit for recommending repairs. If the case-based reasoning tool determines that the probability that a specific repair will resolve a fault is above a threshold probability value, then that repair (in the form of a repair code) will be reported by the case-based reasoning tool 299. The case-based reasoning tool 299 prioritizes the repair recommendations and reports the top five repair codes, as depicted by reference character 306. Following processing by the case-based reasoning tool 299, the system will run the case repetition detection process (see reference character 118 in Figure 8).

Figure 11 illustrates the Bayesian belief network tool 310. Each version of the Bayesian belief network tool 310 uses a specific rule base, as depicted by reference character 314. The specific configuration selected is based on the locomotive road number. A reference character 316 depicts a table linking causes identified by the Bayesian belief network tool 310 to specific repairs for a problem case. The Bayesian belief network rule base 314 also identifies the repair probability thresholds used for prioritizing repairs. Like the case-based reasoning tool 299, the Bayesian belief

network tool 310 uses a seven day look-back in one embodiment. This look-back is modified (as discussed in conjunction with Figures 9A and 9B) to eliminate the effects of closed or recommended cases. The output from the Bayesian belief network tool 310 is the top three repair codes. After the Bayesian belief network tool 310 runs, the system runs the case repetition detection tool as illustrated by reference character 118 in Figure 8.

Figure 12 illustrates the fault classification tool 326. This tool receives input from the fault log of the current download, just as the tools discussed previously, as shown by a reference character 328. There is no look-back period associated with execution of the fault classification tool 326. Also input to the fault classification tool 326 is a fault service strategy table 330. This table comprises a list of typical faults found within a railroad locomotive and a priority ranking for each. Each fault in the table is identified with an indicator value as either a "critical fault", "other fault", or "not found on the fault service strategy table". The fault classification tool compares the faults from the fault log 328 with those listed in the fault service strategy table 330, to assign an indicator value to each fault. The output fault codes with the indicator value are depicted by a reference character 332 in Figure 12.

Figure 13 illustrates the data pack anomaly detection (DPAD) tool 336. This tool operates on fault log operational parametric data (also referred to as "data pack" data) (see reference character 338) within the look-back period. The data pack data is collected when a fault occurs and provides a measure of operational conditions (voltage, temperature, etc.) of selected locomotive systems. The DPAD rules are programmed into the data pack anomaly detection tool 336, and the data pack anomaly detection tool 336 is configured, using the locomotive road number, by parameters in the configuration table 30. The "data pack" consists of 16 parameters (in one embodiment) that are sampled upon the occurrence of each fault. The data pack anomaly detection tool examines the parametric values and the accompanying fault to determine a repair recommendation. The output from the data pack anomaly detection tool 336 is a list of repair codes including all repair codes that are indicated by the rule comparison process. The output repair codes are depicted generally in Figure 13 by reference character 344.

The anomaly detection tool 350 is illustrated in Figure 14. This tool analyzes parameters received from the current download case (see reference character 352) and compares the parameters with limits and criteria from the anomaly definitions. A diagnostic engine map file 354 supplies internal anomaly codes that are mapped to parameters in the anomaly definition table. Thus, when a particular parameter correlates with an anomaly in the table, the anomaly detection tool outputs the internal code associated with that anomaly. Configuration data for the anomaly detection tool 350 is input from an initiation file stored in the configuration table 30. This file provides the initial configuration data, including the anomaly detection tool version number that is to execute based on the locomotive road number from which downloaded parametric performance data was collected. The anomaly indicators provided as an output by the anomaly detection tool 350 are indicated by reference character 360. In addition to the anomaly indicators 360, the anomaly detection tool 350 provides derived parameters (for example, statistics) as an output. These are indicated in Figure 14 by reference character 362. These derived parameters are calculated from parametric performance data in the download case and are saved to a database or table for use in graphs and other analysis aids.

Figure 15 illustrates the trend anomaly tool 370. Like the anomaly detection tool 350, this tool also compares specific operational parameters from the locomotive with values defined in the anomaly definition table, represented generally by reference character 372. Configuration information is provided from the configuration table 30 for identifying the specific version of the trend anomaly tool 370 that is to operate on the data, based on the locomotive road number. Parametric performance parameters uploaded from the locomotive (and illustrated by reference character 376) are input to the trending tool 370. Only the current download case information is used by the trend anomaly tool 370. Also input to the trend anomaly tool 370 is a state file 378, which includes statistical data (e.g., mean, median, standard deviation) derived from historical performance data. The trend anomaly tool 370 analyzes the current parameter data against the historical statistics and compares the results of this analysis with limits and criteria set forth in the anomaly definitions, as provided by the definition table 372. The trend anomaly tool 370 outputs the anomaly identifiers associated with the results of this comparison process (see

reference character 380) and updates the statistics contained within the state file, as indicated by reference character 382. The state file is re-initialized if there are any closed or recommended cases within the look-back period. Also output from the trend anomaly tool 370 are derived parameters 384, which are useful for creating
5 graphs, charts and other analysis aids. As discussed in conjunction with the other tools that are run, following execution of the trend anomaly tool 370, a case repetition detection program is run (as illustrated by reference character 118 in Figure 8).

To focus limited resources on solving only new unreported problems, it is necessary to avoid the creation of new problem cases when existing cases cover the
10 same previously reported problem. The features of the case repetition detection element include: the ability to distinguish a new problem case based upon the makeup of detected faults, anomalous conditions, and recommended repairs reported by the automated analysis tools of the present invention; the ability to create a new
15 problem case to store information about a new problem, the ability to maintain an open time frame so that related data can be analyzed and combined into a single problem case if necessary; and the ability to distinguish and link additional relevant data to pre-existing cases instead of creating a new case.

Returning to Figures 9A and 9B, the case repetition detection element of the present invention is shown within the dash lines identified by reference character 420.
20 The result of the case repetition detection process is the creation of a new case (see the steps 224 and 238) or the addition of the current anomaly and fault information to an existing case, as depicted at the step 232.

The case repetition detection process is also shown diagrammatically in Figure 16. Reference characters 422 and 424 depict input values to the case repetition
25 process 420. The input value represented by reference character 422 is the number of hours after a problem case is created during which all tool outputs should be combined into a single case (see reference character 422), rather than creating multiple cases. This input value is user defined and referred to as "x" in the decision
step 230 of Figure 9A. To run the case repetition detection process, current repairs,
30 faults, and anomalies identified by the other analysis tools are used as input values (see reference character 424 of Figure 16). If there are no problem cases within the selected combination period, then a new problem case will be created. If there is a

problem case within the combination period, then all the repair recommendations made during that period (including the current recommended repairs) are combined into one problem case. As discussed above, each case includes the faults and anomalies associated with the repair recommendation and therefore this information is also contained within the problem case. If processing is outside the case combination period, the case repetition detection process 420 checks all the open problem cases outside the case combination period and attaches the new problem case as a child to an existing problem case if the repairs of the two problem cases match and if the list of anomalies or faults in the new problem case are contained in the existing problem case. This feature is also depicted at the step 232 and 242 of Figures 9A and 9B. If there is no match, then a new problem case is created. The creation of a new case by the case repetition detection process 420 is depicted at an output step 426.

Another important feature of the present invention is the re-analysis of the created problem cases after the completion of a recommended repair. This process is shown in Figures 9A and 9B by reference character 440. This aspect of the present invention is implemented by the use of a look-back parameter as previously discussed herein. The objective of this feature is to screen out anomalies or faults that in fact have already been addressed through recent repair actions or recommendations. Following is a summary of the steps involved in the re-analysis process 440. Repairs are not added to the list if all of the following conditions are met: the results of the analysis indicates that there is an anomalous condition and/or repair code needed (see the decision step 166 of Figure 9A); the locomotive has been repaired or repair recommendations have been made recently (see the decision step 170 of Figure 9A); the anomalous conditions and/or repair codes are the same as those that were identified before the repair recommendation or operation (see the decision step 180 of Figure 9A); the data that indicated an anomalous condition or repair is re-analyzed so that input download data preceding the repair recommendation or operation is not included within that re-analysis (see the step 182 of Figure 9A; and the re-analysis indicates that no anomalous condition is present and no repair is needed (see the step 184 and the decision step 186 of Figure 9A).

Figure 17 illustrates the process for identifying certain faults as critical faults and for indicating this in the fault service strategy table 330. Further, the on-board

monitor communicates fault information for analysis according to the analysis process set forth herein based on either a specific time schedule or according to the severity of the particular fault encountered. In particular it is required to have a process for determining the so-called "critical faults," as defined below.

5 Figure 17 illustrates an exemplary flow chart of a process for identifying malfunctions, e.g., faults and/or operational parameters, that are indicative of impending locomotive road failures. Upon start of operations at step 500, at a step 502 all faults logged for a predetermined time interval, e.g., the last 12 months or any other selected time interval are retrieved. This step is accomplished by reviewing, for
10 instance, the problem cases created at the step 102 of Figure 8. At a step 504, the process identifies faults that occur relatively frequently. Step 506 allows for identifying the number of locomotives that are affected the most by the frequently occurring faults. For example, as shown in Table 1 below, fault code 1000 occurs 1306 times over a predetermined time interval, fault code 1001 occurs 500 times over
15 the same time interval, and fault code 1002 occurs 1269 times over the same time interval. As further shown in Table 1, although fault code 1002 occurs more frequently relative to fault code 1001, since the number of locomotives affected by fault code 1001 is larger compared to the number of locomotives affected by fault code 1002, then the relative ranking of fault code 1001 in terms of fleet percentage
20 affected, is higher for fault code 1001 than for fault code 1002. At a step 508, the faults are classified into various types of faults, e.g., critical, restrictive, non-restrictive, special interest, etc. As used herein, a critical fault is a malfunction indication that would indicate imminent complete loss of locomotive power, potential damage to the failing subsystem and/or locomotive, or safety issues. A restrictive
25 fault is a malfunction indication that would prevent the locomotive from operating at full power or performance due to, for example, mechanical, electrical and/or traction power malfunctions. A special interest fault may be related to a customized field experiment or analysis undertaken for a railroad operator or project, may be used for monitoring trending of predetermined operational parameters, etc.

30

Fault	No. of Occurrences	No. of Locomotives	Percentage of Fleet
1000	1306	102	39%
1001	500	83	32%
1002	1269	80	31%
1003	541	70	27%

Table 1

Step 512 allows for conducting expert analysis or review by expert personnel, e.g., MDSC personnel and/or engineering teams responsible for servicing any affected subsystems, e.g., traction motors, fuel delivery subsystem, etc.

As suggested above, step 514 allows for processing, if desired, of special interest faults, failure trends, critical faults, etc. In particular, the analysis scheduler 15 (see Figure 2) can be used to batch process any group of fault data in accordance with the teachings of the present invention. The step 516 allows for storing, in a suitable database, every fault that would trigger a respective locomotive to make a call home request. In one embodiment these are classified as the call home or critical faults. As shown at step 518, the process is an iterative process that may be repeated so as to maintain an up-to-date database of call home faults. The updating may be performed at predetermined time intervals, or may be performed due to special events, such as deployment of new models of locomotives, locomotive upgrades, etc.

WHAT IS CLAIMED IS:

1. A method for scheduling the execution of one or more analysis tools (110, 112, 114, 116, 128, 130) operating on performance data of a plurality of mobile assets, (2) to assess the need for remedial action to one or more of the mobile assets
5 (2), comprising:
 - a) receiving the performance data; (106)
 - b) storing the performance data;
 - c) selecting the highest priority unanalyzed performance data; (61)
 - d) establishing a limit on the number of executions available to be
10 performed during a predetermined time interval for each of the one or more analysis tools; (62)
 - e) providing the selected unanalyzed performance data to one or more of the analysis tools if the execution limit for that tool has not been reached; (63) and
 - f) generating a mobile asset specific recommendation based on the results
15 derived from the one or more analysis tools. (306, 318, 332, 344, 360)
2. The method of claim 1 wherein the mobile asset is a railroad locomotive. (2)
3. The method of claim 1 wherein the performance data comprises parametric performance data for the mobile asset.
- 20 4. The method of claim 1 wherein the performance data comprises fault data for the mobile asset.
5. The method of claim 1 wherein the performance data comprises data indicating the occurrence of an out-of-spec condition on the mobile asset.
6. The method of claim 1 wherein the mobile asset (2) specific
25 recommendation comprises a repair recommendation. (306, 318, 344)
7. The method of claim 1 wherein the mobile asset (2) specific recommendation comprises a request to collect additional performance data from the mobile asset. (2)
8. The method of claim 1 wherein the mobile asset (2) specific
30 recommendation comprises a request for review of the performance data by an expert in the operation and repair of the mobile asset. (224, 238)
9. The method of claim 1 wherein the step c) comprises:

- c1) creating a plurality of performance data sets, (21) wherein each set includes performance data related to an operational event aboard the mobile asset;
- c2) assigning a priority ranking to each performance data set; and
- c3) selecting the highest priority performance data set therefrom. (44)

5 10. The method of claim 1 wherein the step c) comprises:

- c1) segregating the performance data into high-priority data and normal-priority data; and
- c2) selecting the highest priority performance data from each of the high-priority data and the normal-priority data. (61)

10 11. The method of claim 10 wherein the step d) comprises:

- d1) establishing a limit on the number of high-priority executions available to be performed during a predetermined time interval for each analysis tool; (30) and
- d2) establishing a limit on the number of normal-priority executions available to be performed during a predetermined time interval for each analysis tool.

15 (30)

 12. The method of claim 1 wherein each of the one or more analysis tools processes the performance data in parallel. (110, 112, 114, 116, 128, 130)

 13. The method of claim 1 further comprising:

20 f) establishing a look-back period, wherein the performance data includes the mobile asset performance data during the look-back period; (182)

 g) determining whether the current mobile asset specific recommendation is substantially similar to one or more previous recommendations;

25 h) if the current mobile asset specific recommendation is substantially similar to at least one previous recommendation within the look-back period, modifying the look-back period such that the modified look-back period begins after implementation of the previous recommendation; (182) and

 i) selecting unanalyzed performance data during the modified look-back period for analysis. (184)

 14. The method of claim 1 further comprising:

30 (f) establishing a combining period for the mobile asset (2) specific recommendations;

g) determining whether there are any open mobile asset specific recommendations during the combining period; (230)

h) if there is at least one open mobile asset specific recommendation during the combining period, including the current recommendation with the open recommendation; (232)

i) if there is not at least one open mobile asset specific recommendation during the combining period, analyzing the current mobile asset specific recommendation from the one or more analysis tools to determine whether the current recommendation is substantially similar to an open recommendation; (244) and

j) if there is a substantial similarity, combining the current mobile asset specific recommendation with the substantially similar open recommendation. (242)

15. An article of manufacture comprising:

a computer program product comprising a computer-usable medium having a computer-readable code therein for scheduling the execution of one or more analysis tools operating on performance data of a plurality of mobile assets, to assess the need for remedial action to one or more of the mobile assets, the computer-readable code in the article of manufacturer comprising:

a computer-readable program code module for storing the performance data;

a computer-readable program code module for selecting the highest priority unanalyzed data; (61)

a computer-readable program code module for establishing a limit on the number of executions available to be performed during a predetermined time interval for each of the one or more analysis tools; (62)

a computer-readable program code module for providing the selected unanalyzed performance data to one or more of the analysis tools if the execution limit for that tool has not been reached; (63) and

a computer-readable program code module for generating a mobile asset specific recommendation based on the results derived from the one or more analysis tools. (306, 318, 332, 344, 360)

16. The computer program product of claim 15 wherein the mobile asset is a railroad locomotive. (2)

17. The computer program product of claim 15 wherein the computer-readable program code module for storing the performance data further comprises:

a computer-readable program code module for segregating the performance data into high-priority data and normal-priority data;

5 a computer-readable program code module for selecting the highest priority performance data from each of the high-priority data and normal-priority data; (61) and

wherein the computer-readable program code module for selecting the highest priority unanalyzed performance data further comprises:

10 a computer-readable program code module for establishing a limit on the number of the high-priority executions available to be performed during a predetermined time interval for each analysis tool; (30) and

a computer-readable program code module for establishing a limit on the number of normal-priority executions available to be performed during a
15 predetermined time interval for each analysis tool. (30)

18. The computer product of claim 15 further comprising:

a computer-readable program code module for establishing a look-back period, wherein the selected performance data includes the mobile asset performance data during the look-back period; (182)

20 a computer-readable program code module for determining whether the current mobile asset specific recommendation is substantially similar to one or more previous recommendations; and

if the current mobile asset specific recommendation is similar to at least one previous recommendation, modifying the look-back period such that the modified
25 look-back period begins after implementation of the previous recommendation, wherein the selected unanalyzed data includes mobile asset performance data during the modified look-back period. (182)

19. The computer product of claim 15 further comprising:

a computer-readable program code module for establishing a combining
30 period for the mobile asset (2) specific recommendations;

a computer-readable program code module for determining whether there are any open mobile assets specific recommendations during the combining period; (230)

if there is at least one open mobile asset specific recommendation during the combining period, a computer-readable program code module for including the current recommendation with the open recommendation; (232)

5 if there is not at least one open mobile asset specific recommendation during the combining period, a computer-readable program code module for analyzing the current mobile asset specific recommendation from the one or more analysis tools to determine whether the current recommendation is substantially similar to an open recommendation; (244) and

10 if there is a substantial similarity, combining the current mobile asset recommendation with the substantially similar recommendation. (242)

20. An apparatus for scheduling the execution of one or more analysis tools operating on performance data of a plurality of mobile assets to assess the need for remedial action to one or more of the mobile assets, wherein each analysis tool includes a predetermined limit on the number of executions available to be performed
15 during a predetermined time interval, said apparatus comprising:

a receiving device (20) for receiving the performance data;

a storage device (211) for storing the performance data;

a controller (15) for selecting the highest priority unanalyzed data from said storage device (21) and for providing the selected performance data as an input to one
20 or more of the analysis tools (110, 112, 114, 116, 128, 130) if the number of executions available to be performed during a predetermined time interval for that tool has not been reached; and

a recommendation creation module (186) for creating a mobile asset specific recommendation based on the results from the one or more analysis tools.

25 21. The apparatus of claim 20 wherein the controller (15) segregates the performance data into high-priority data and normal-priority data, and selects the highest priority performance data from each of the high-priority data and the normal-priority data.

30 22. The apparatus of claim 20 including a look-back period wherein the selected performance data describes the operation of the mobile asset during the look-back period;

a comparator for determining whether the current mobile asset specific recommendation is substantially similar to one or more previous recommendations, and if the current recommendation is substantially similar to at least one previous recommendation, (244) modifying the look-back such that said modified look-back begins after implementation of the previous recommendation, wherein the selected performance data describes the mobile asset performance during the modified look-back.

23. The apparatus of claim 20, further comprising:

a case combining period; (230)

10 a processor for determining whether there are any open mobile assets specific recommendations during said combining period, and if there is at least one open recommendation during said combining period, for combining the current mobile assets specific recommendation with the open recommendation; (242)

15 if there is not at least one open mobile assets specific recommendation during the combining period, a comparator responsive to said processor for analyzing the current mobile assets specific recommendation from the one or more tools to determine whether the current recommendation is substantially similar to open recommendations, and if there is a substantial similarity, for combining the current mobile asset specific recommendation with the substantially similarly recommendation. (242)

24. An apparatus (15) for scheduling one or more analysis tools operating on performance data from a mobile asset, said apparatus comprising:

a receiver (20) for receiving the performance data;

a storage device for storing the performance data; (21, 22)

25 a controller (15) for segregating the performance data into high-priority data and normal-priority data;

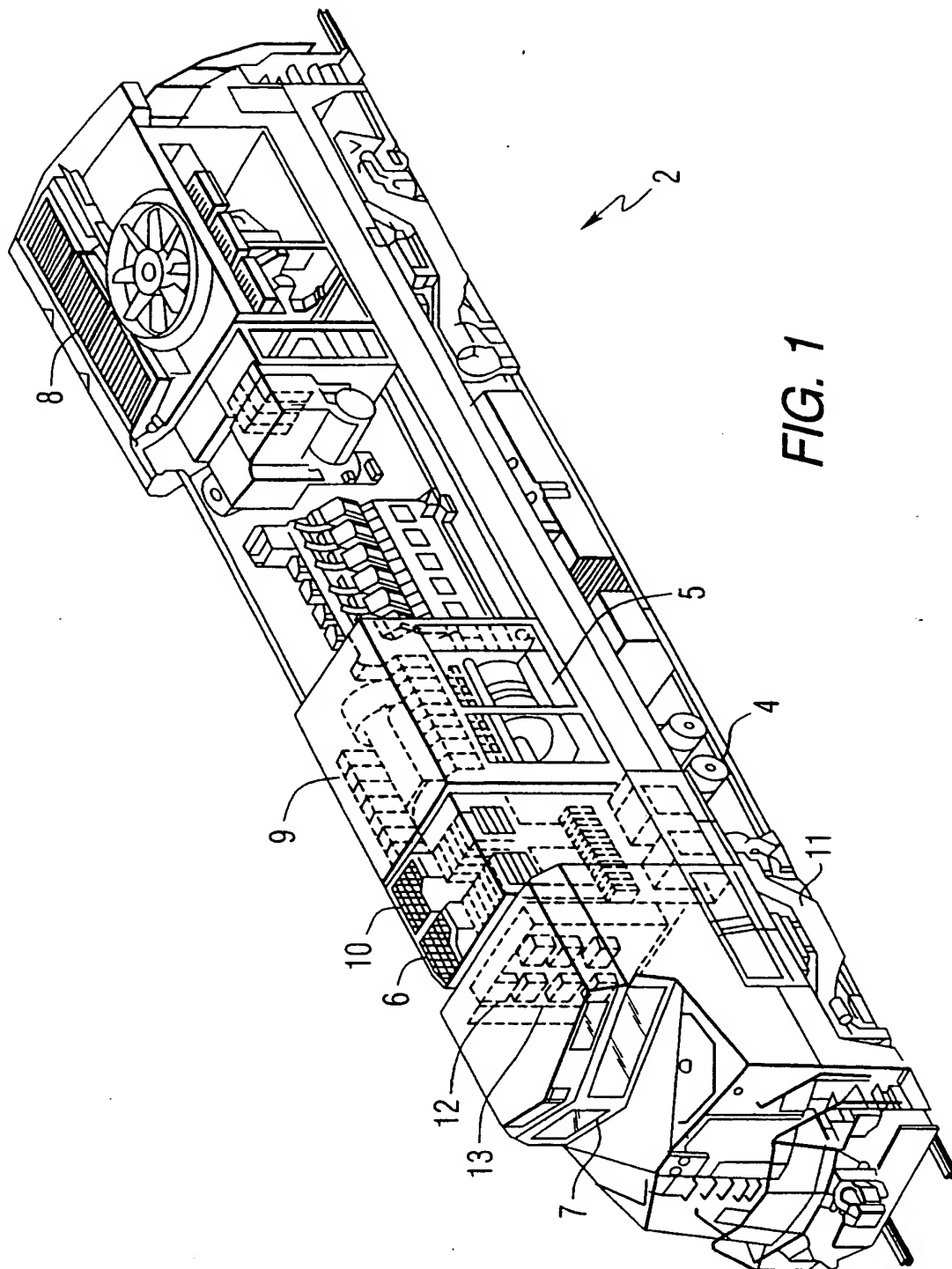
a selector for selecting the highest priority performance data from the high-priority data and the normal-priority data; (44)

30 a first limiter for establishing a limit on the number of high-priority executions for each analysis tool that are available to be performed during a predetermined time interval; and

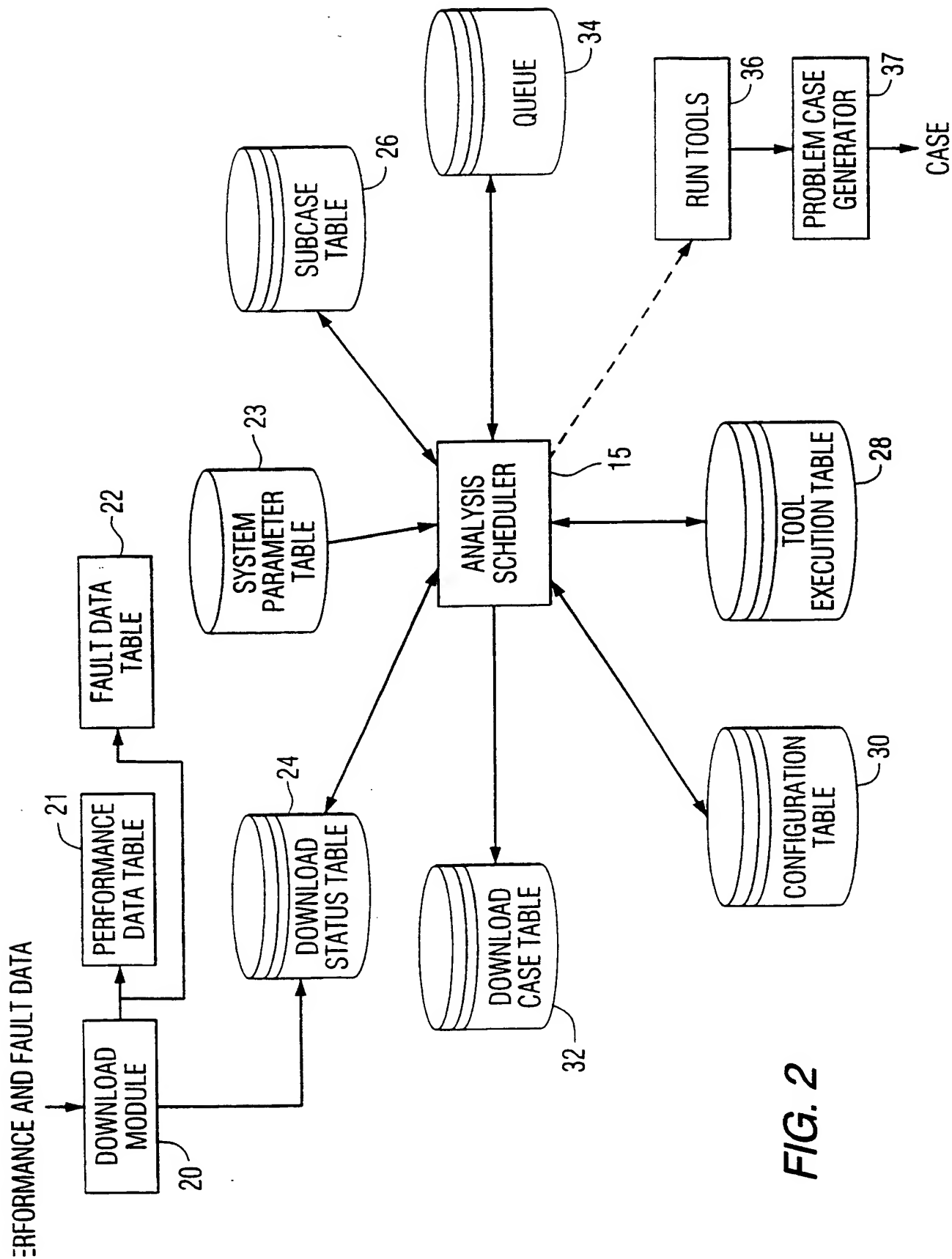
a second limiter for establishing a limit on the number of normal-priority executions for each analysis tool that are available to be performed during a predetermined time interval for each analysis tool.

25. A method for identifying operationally significant faults aboard a mobile asset, said method comprising:
- a) receiving performance data from the mobile asset; (20)
 - b) providing the performance data to one or more analysis tools; (15)
 - c) determining the existence of a fault aboard the mobile asset; (502)
 - d) generating a mobile asset specific recommendation by the one or more analysis tools;
 - e) comparing the mobile asset specific recommendation with a list of recommendations indicating an operationally significant fault; and
 - f) determining that the fault is an operationally significant fault based on the results of step e).
26. The method of claim 25 further comprising:
- g) comparing the fault with a list of operationally significant faults; (504)
- and
- h) determining that the fault is operationally significant based on the results of step g). (508)

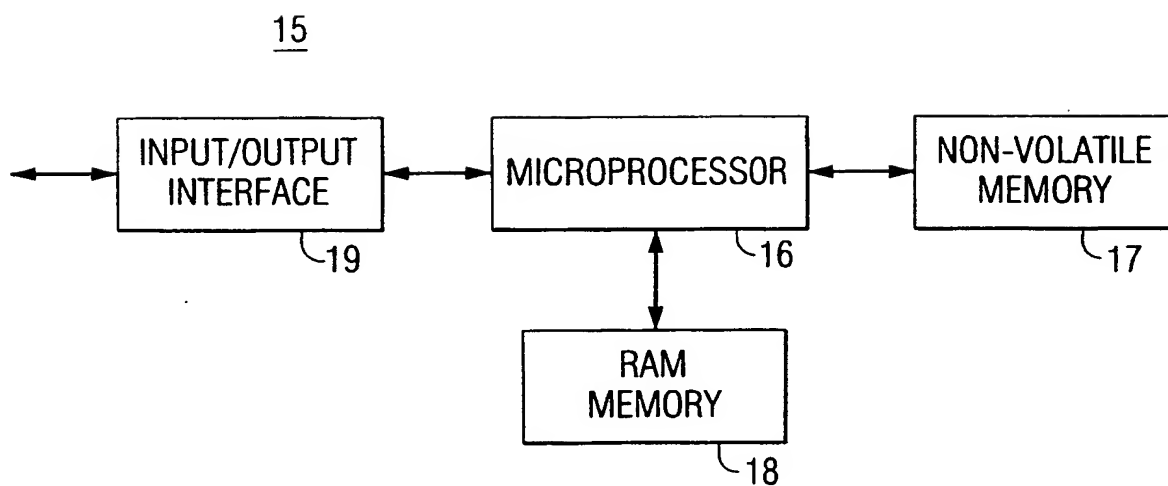
1/19



2/19



3/19

*FIG. 3*

4/19

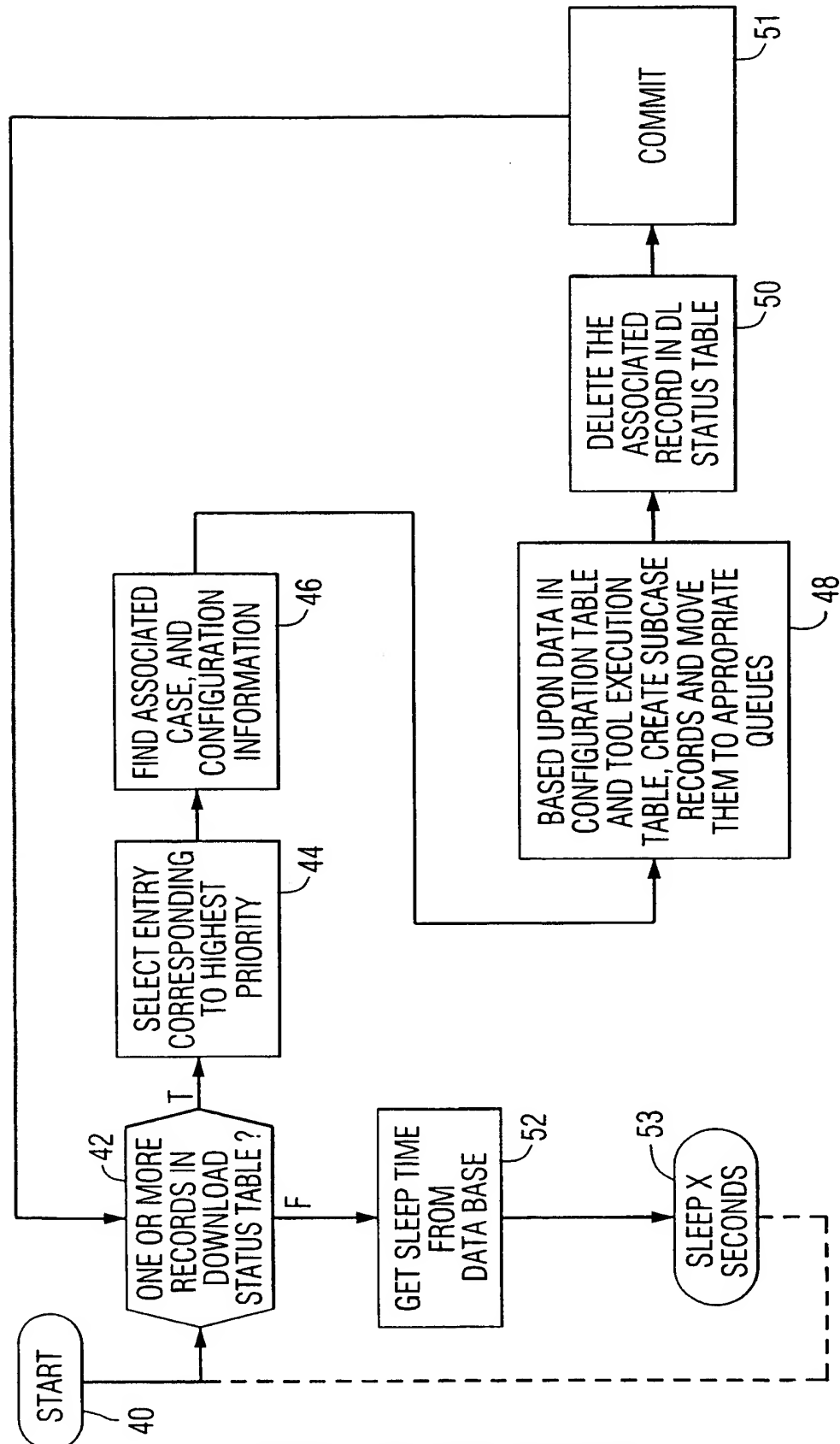


FIG. 4

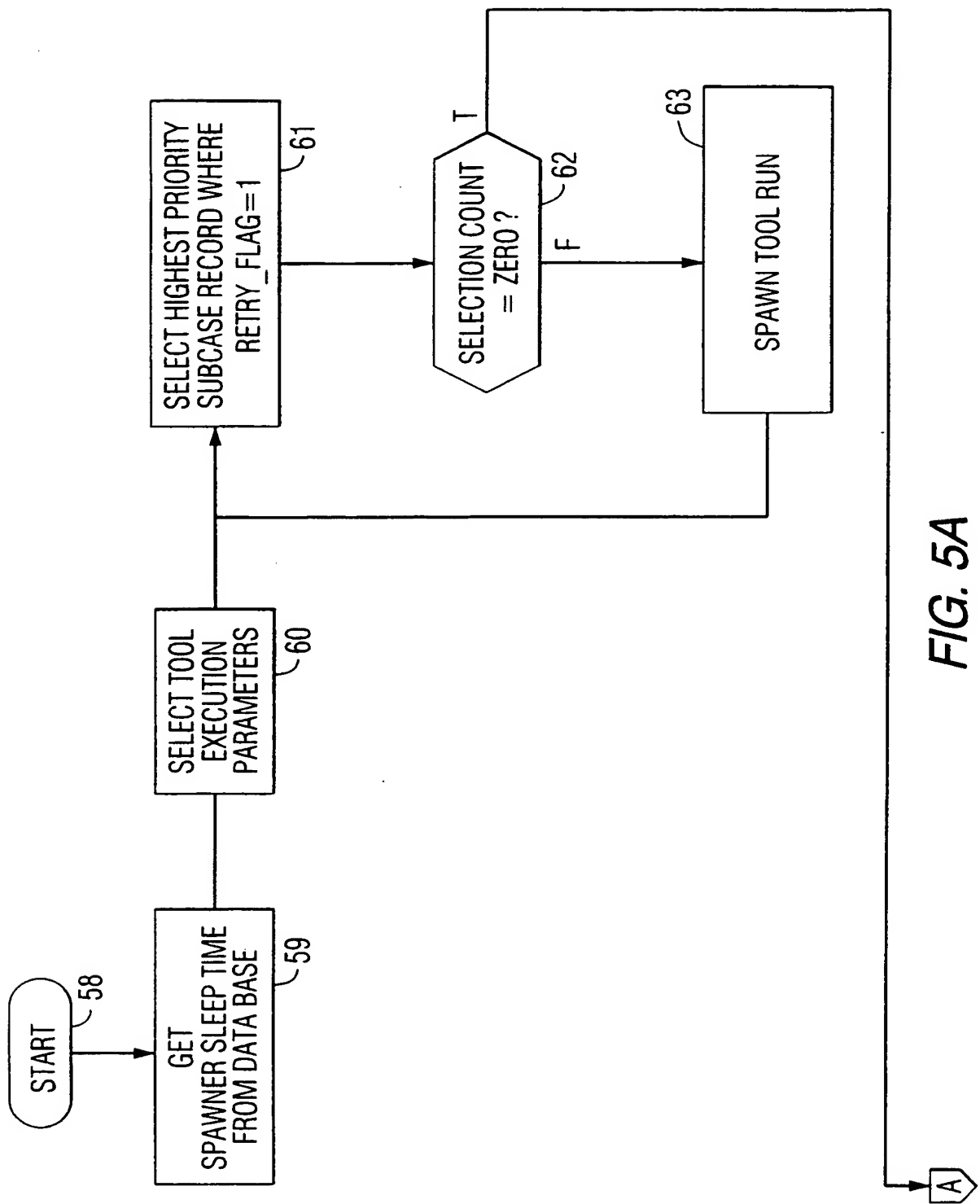
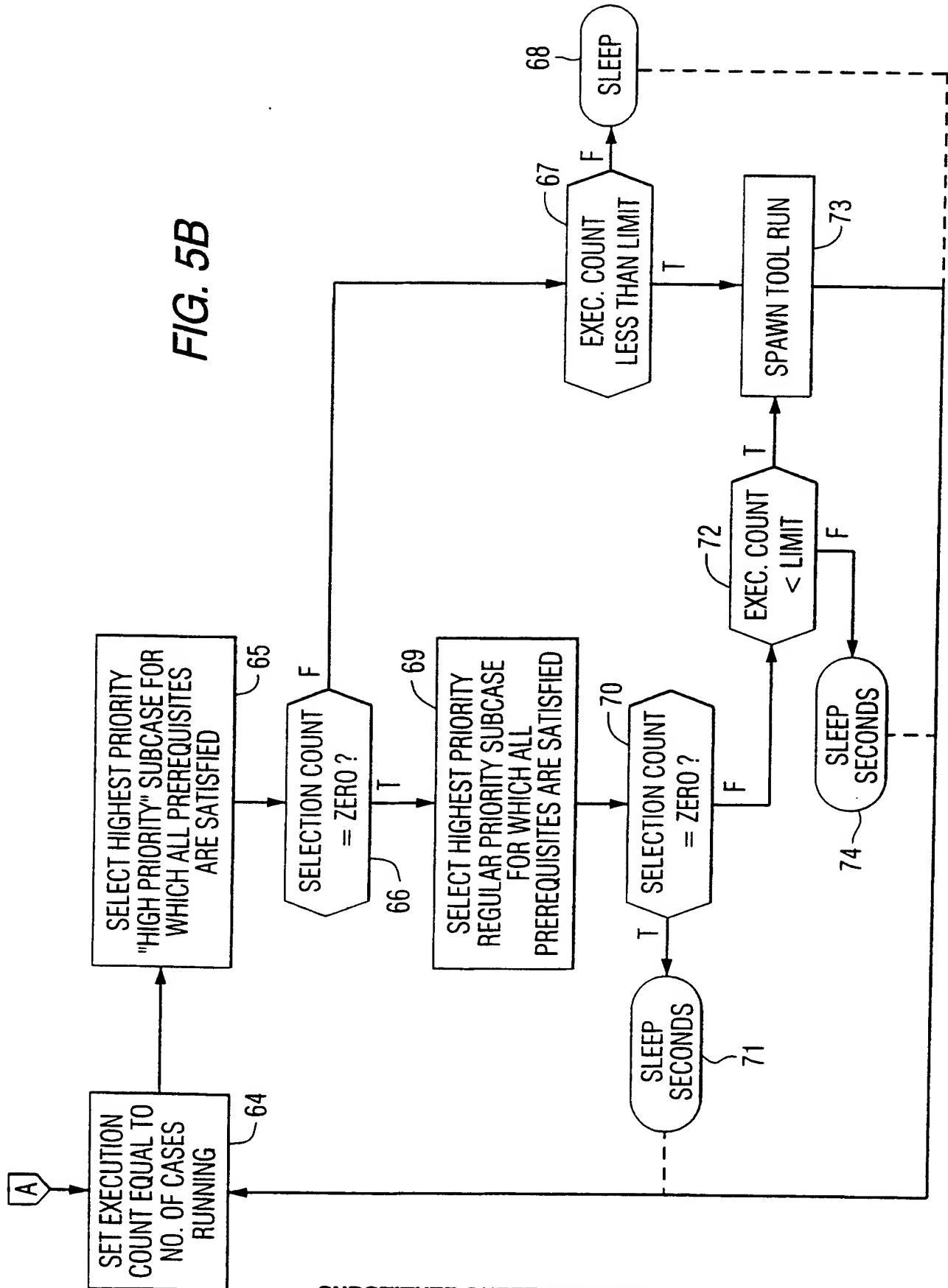


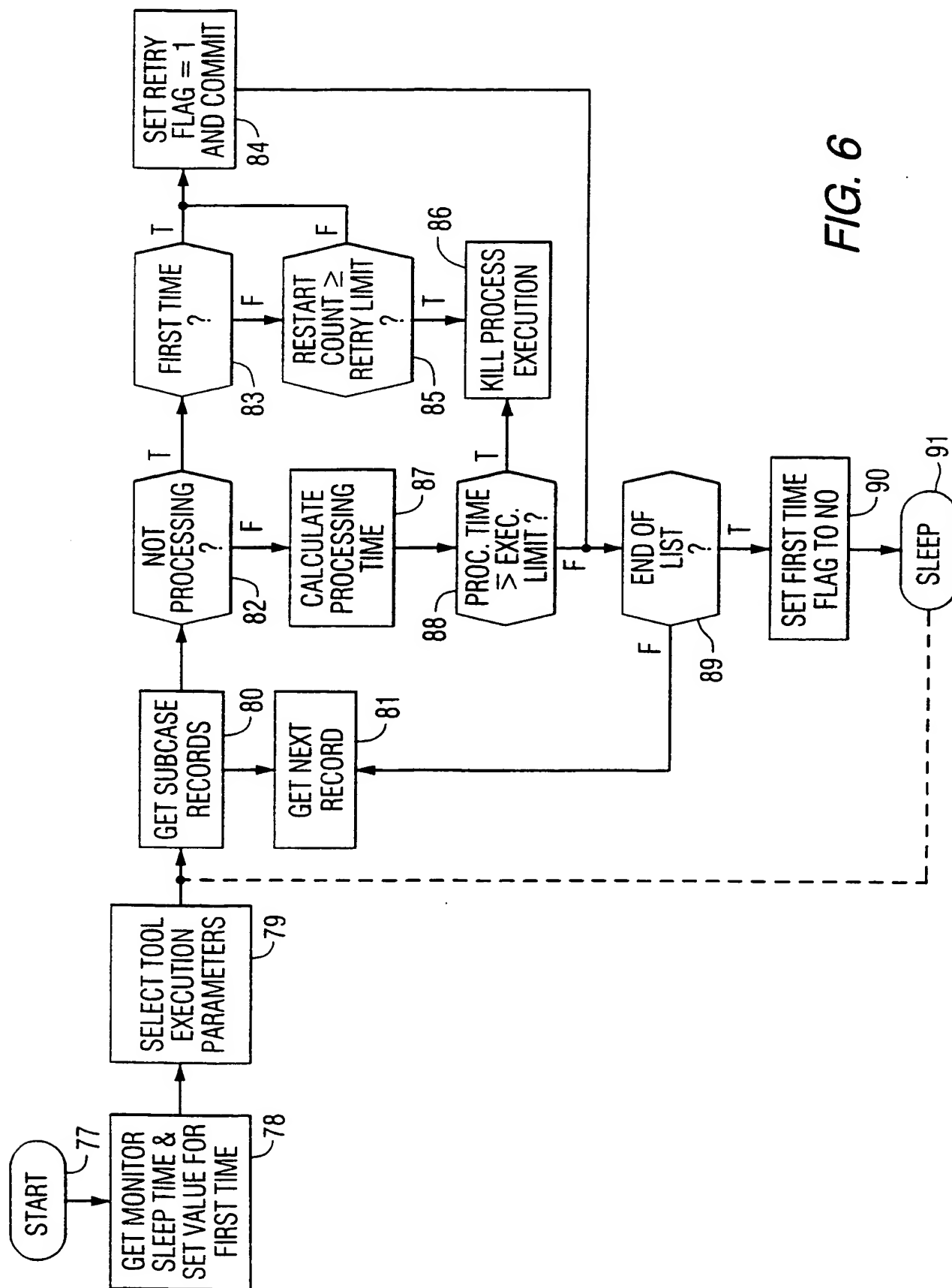
FIG. 5A

6/19

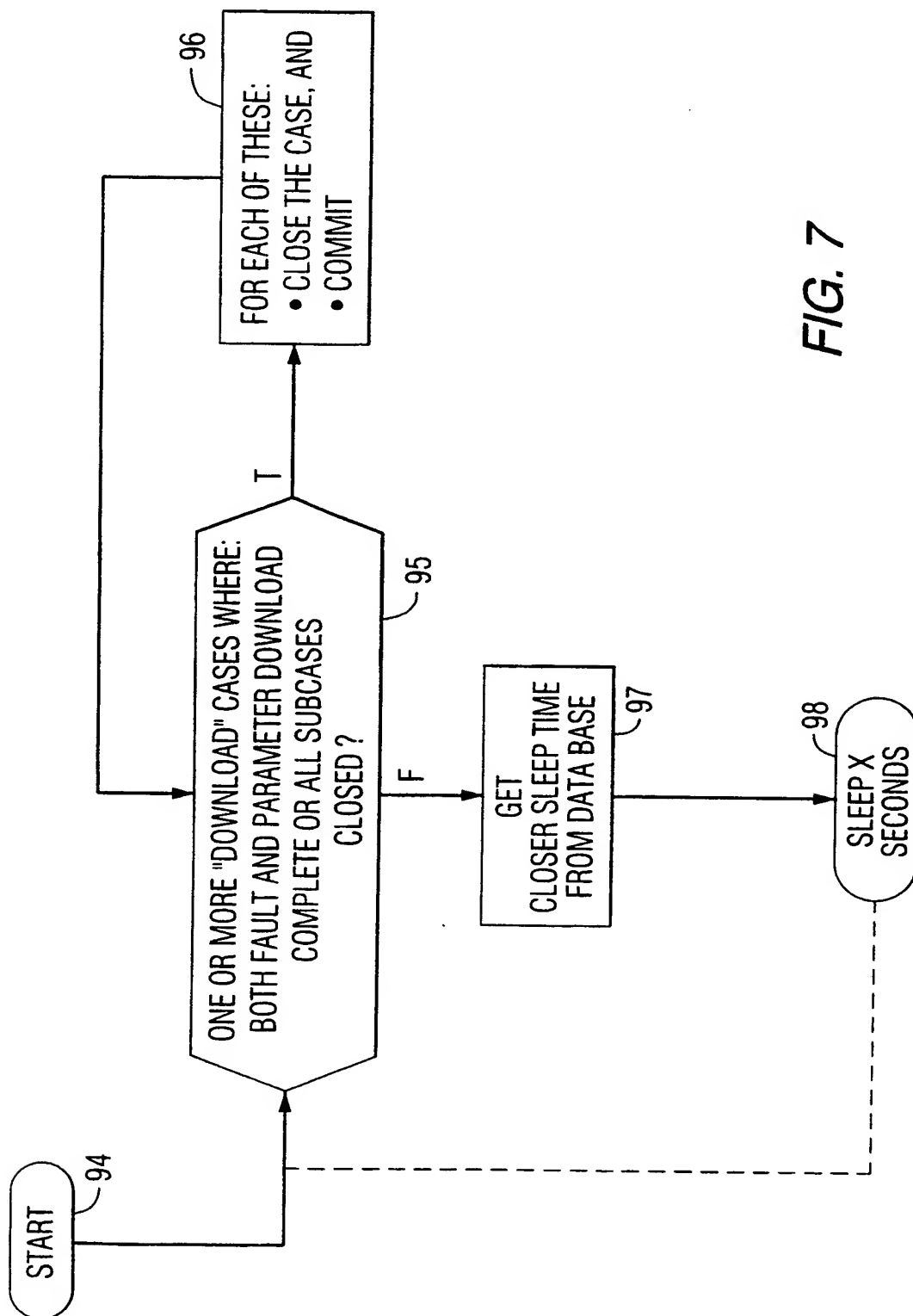
FIG. 5B



7/19



8/19



9/19

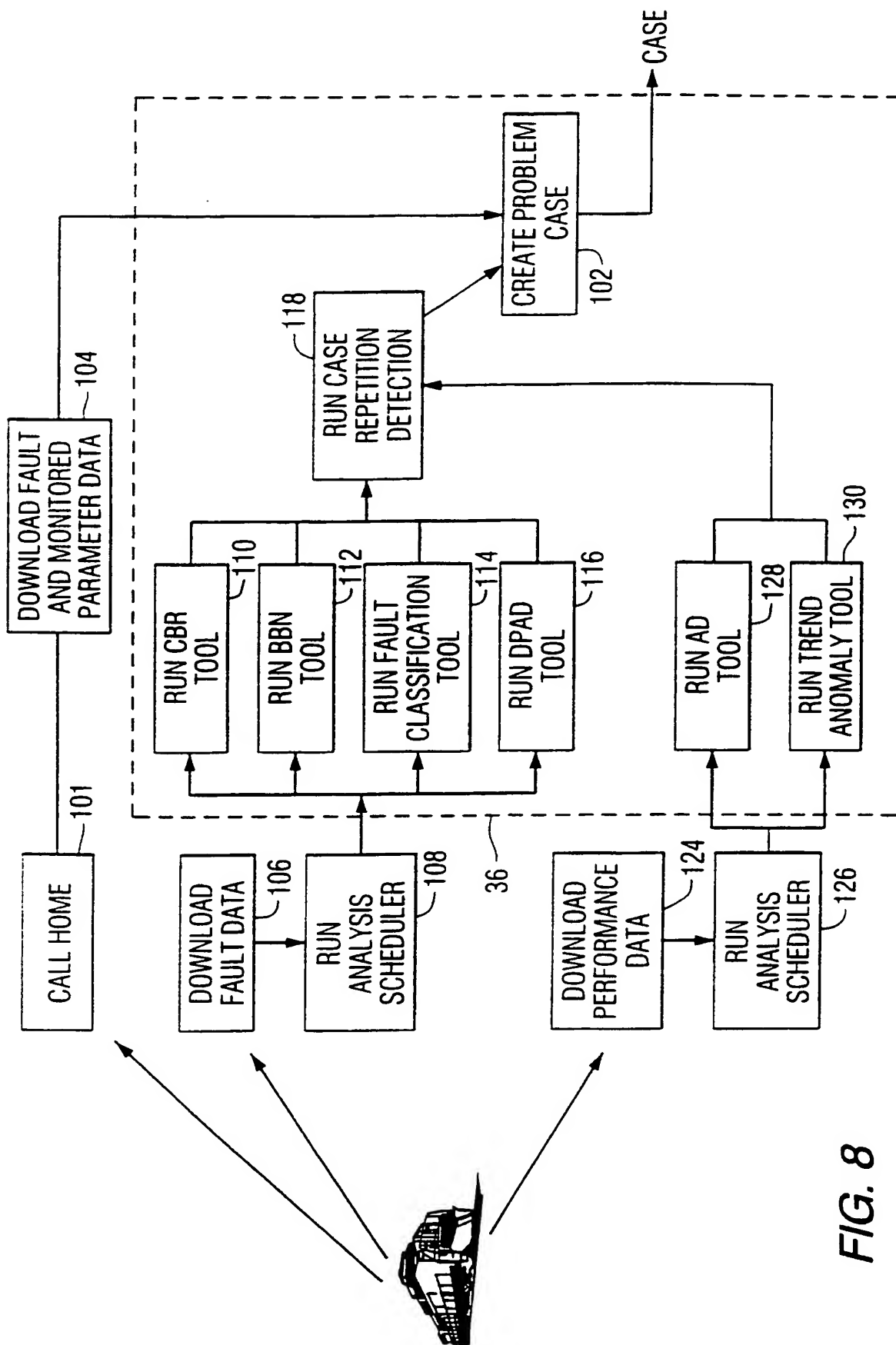


FIG. 8

10/19

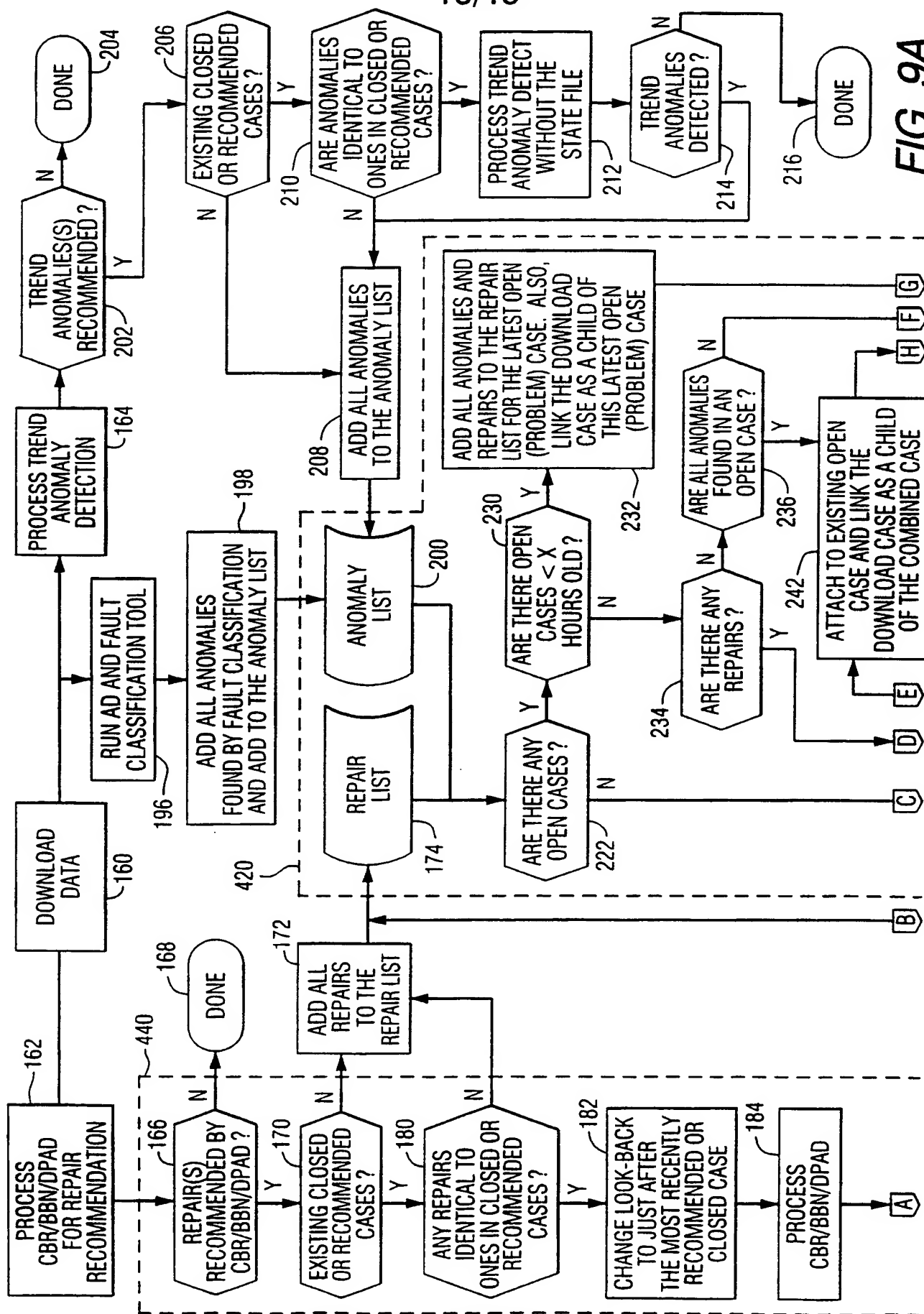
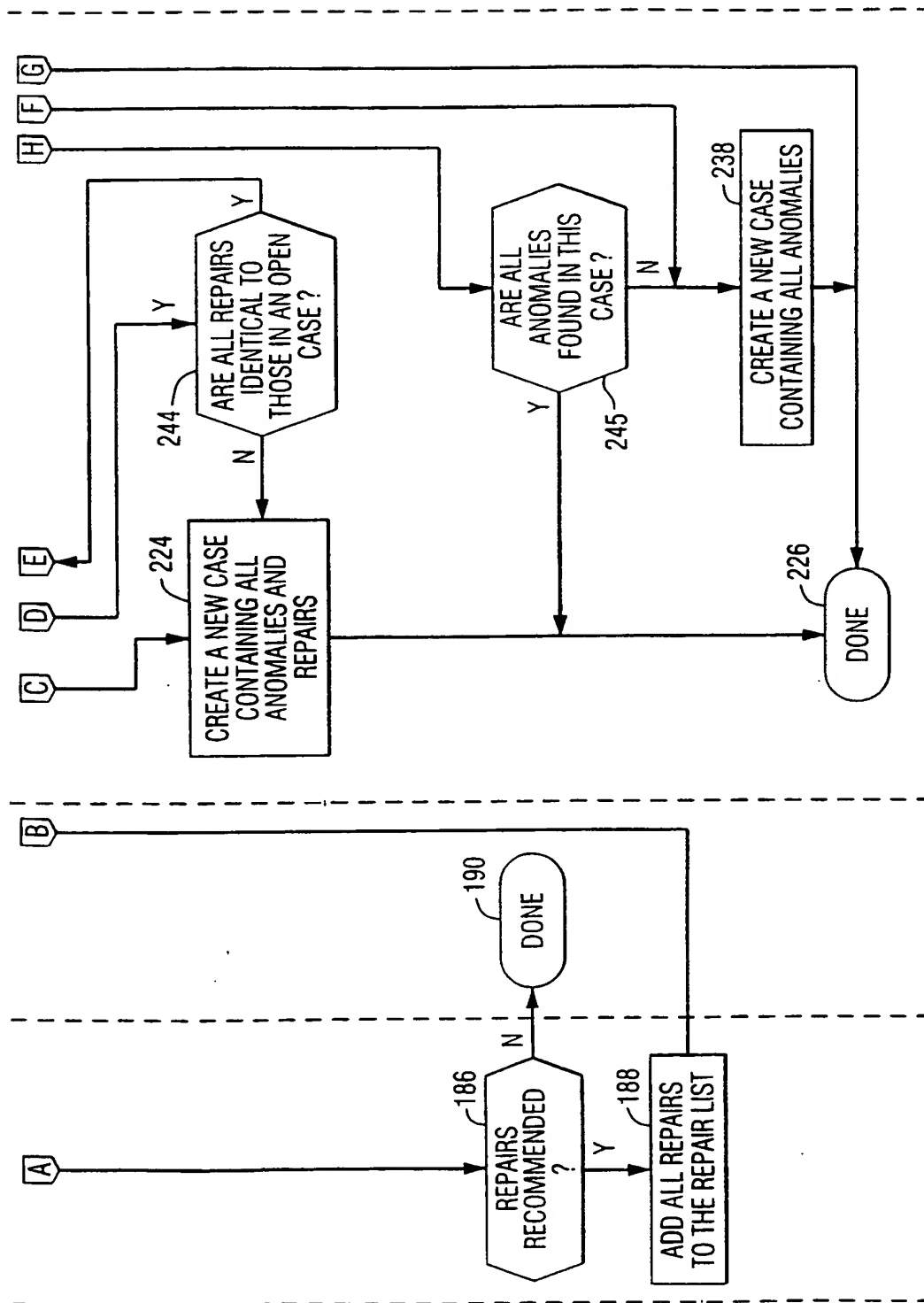


FIG. 9A

11/19

FIG. 9B



12/19

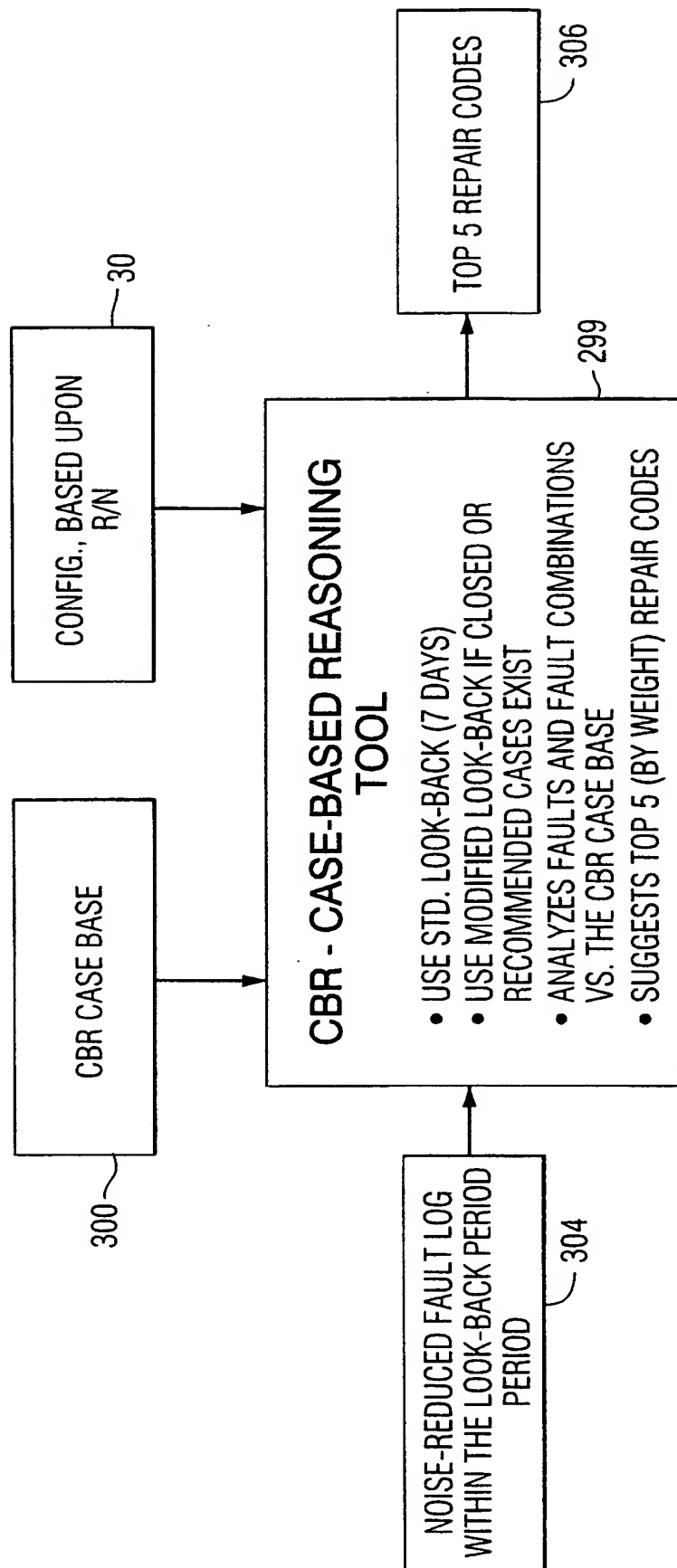


FIG. 10

13/19

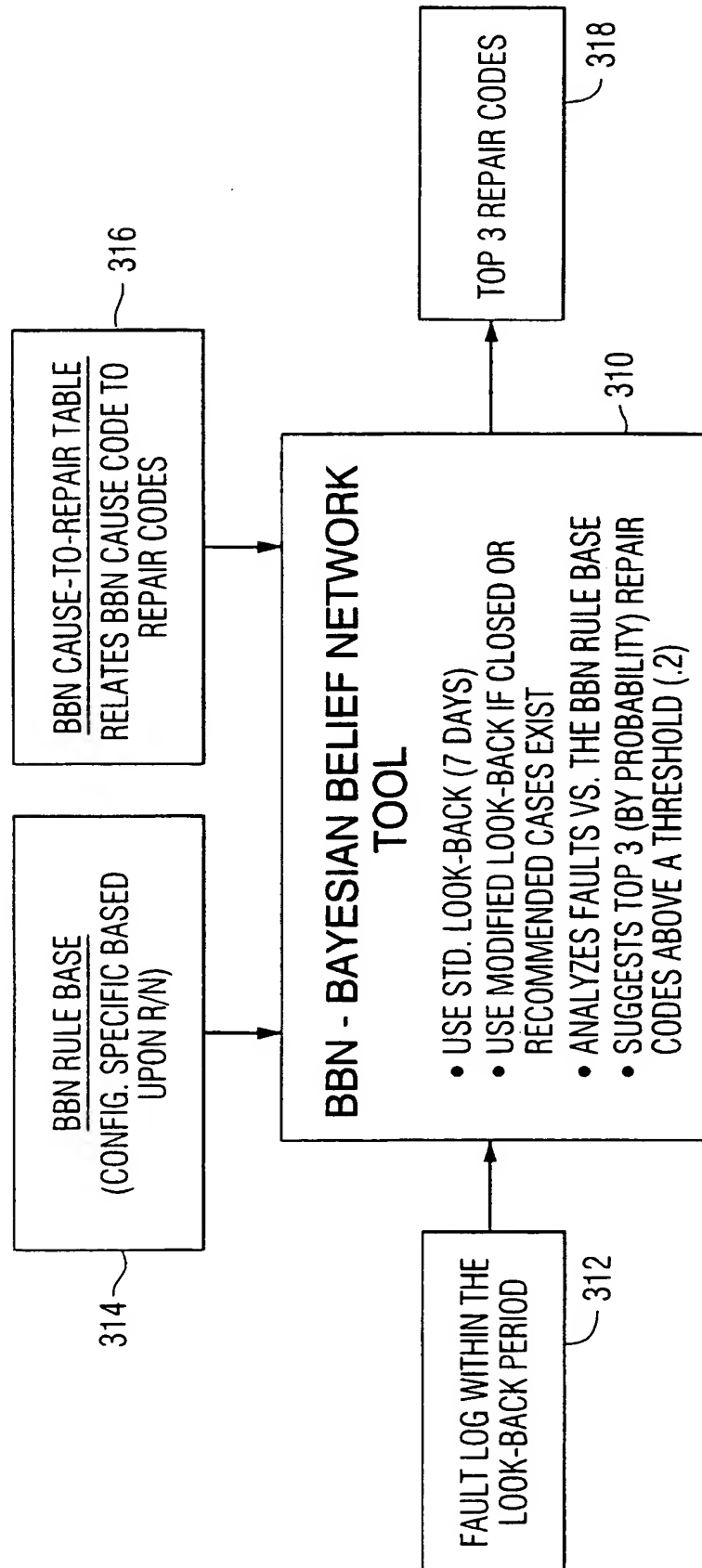


FIG. 11

14/19

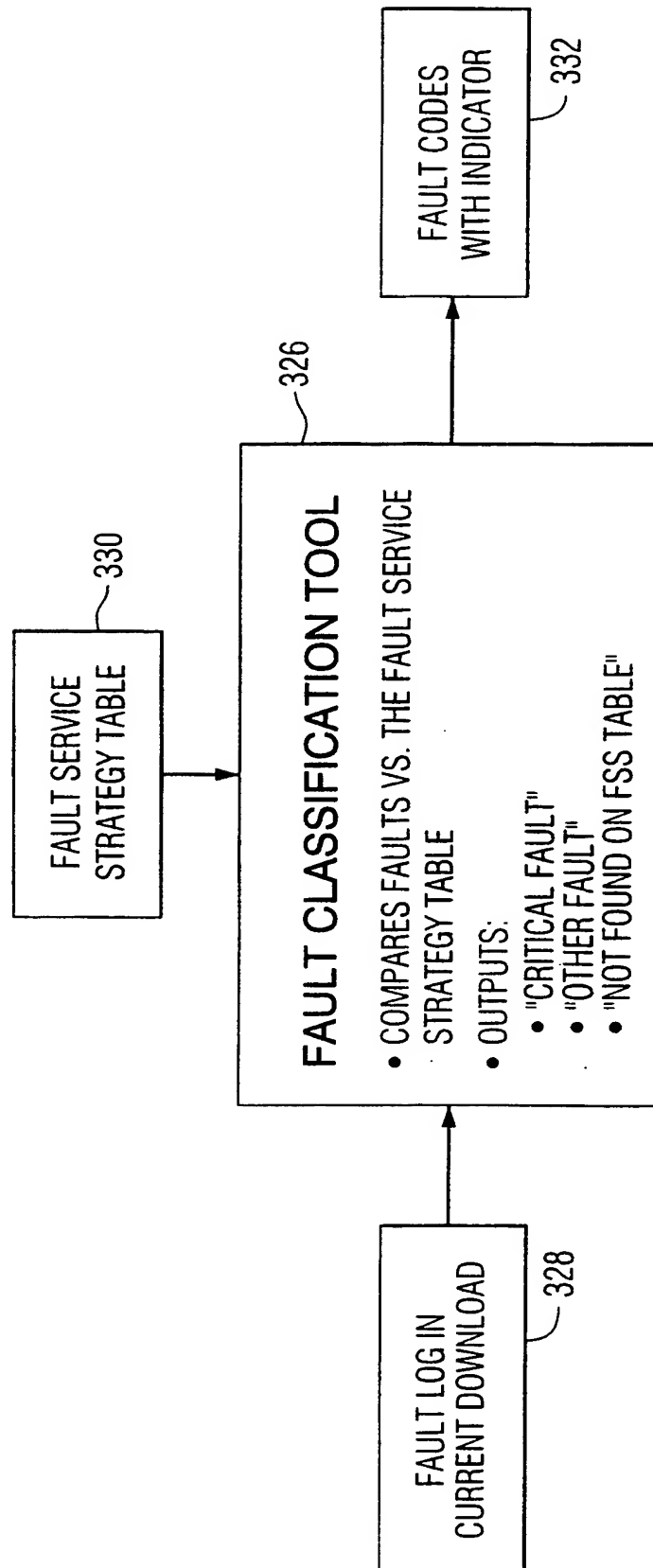


FIG. 12

15/19

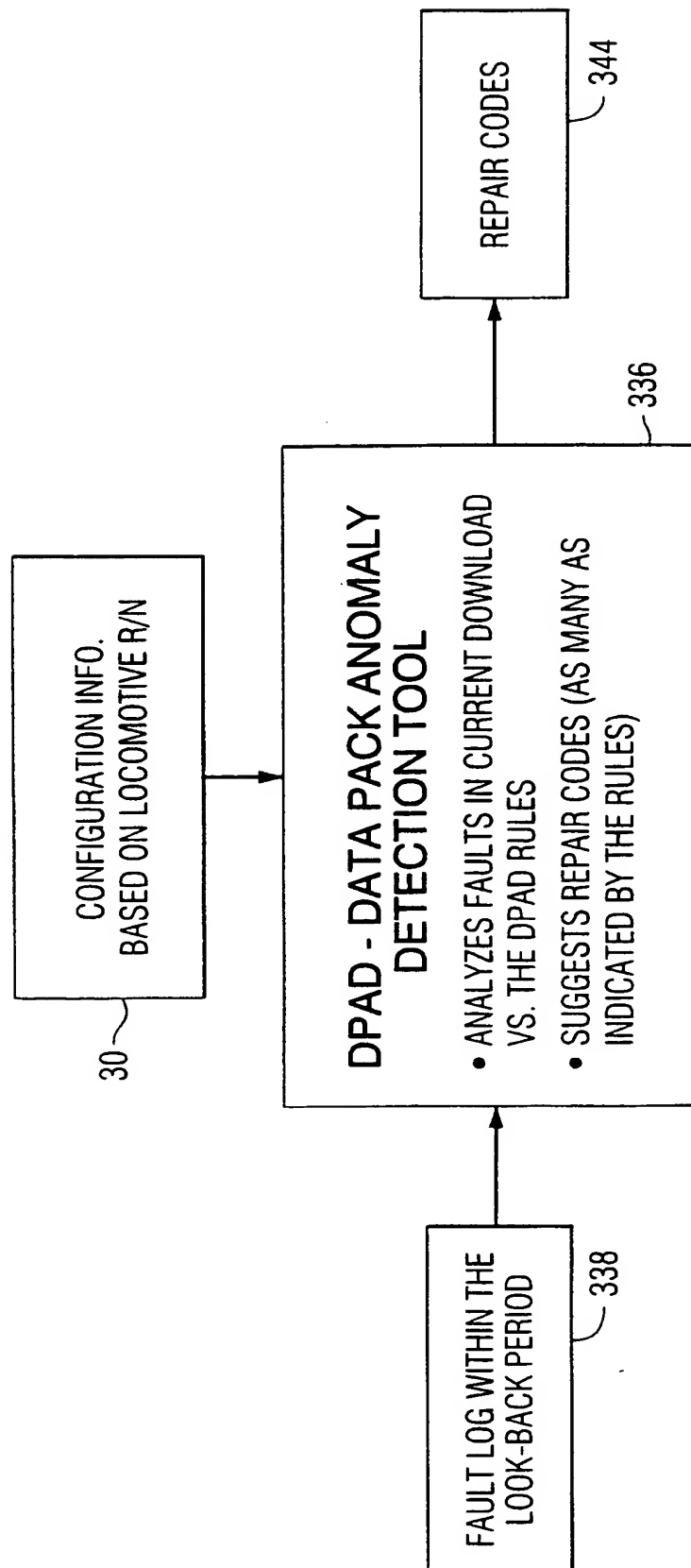


FIG. 13

16/19

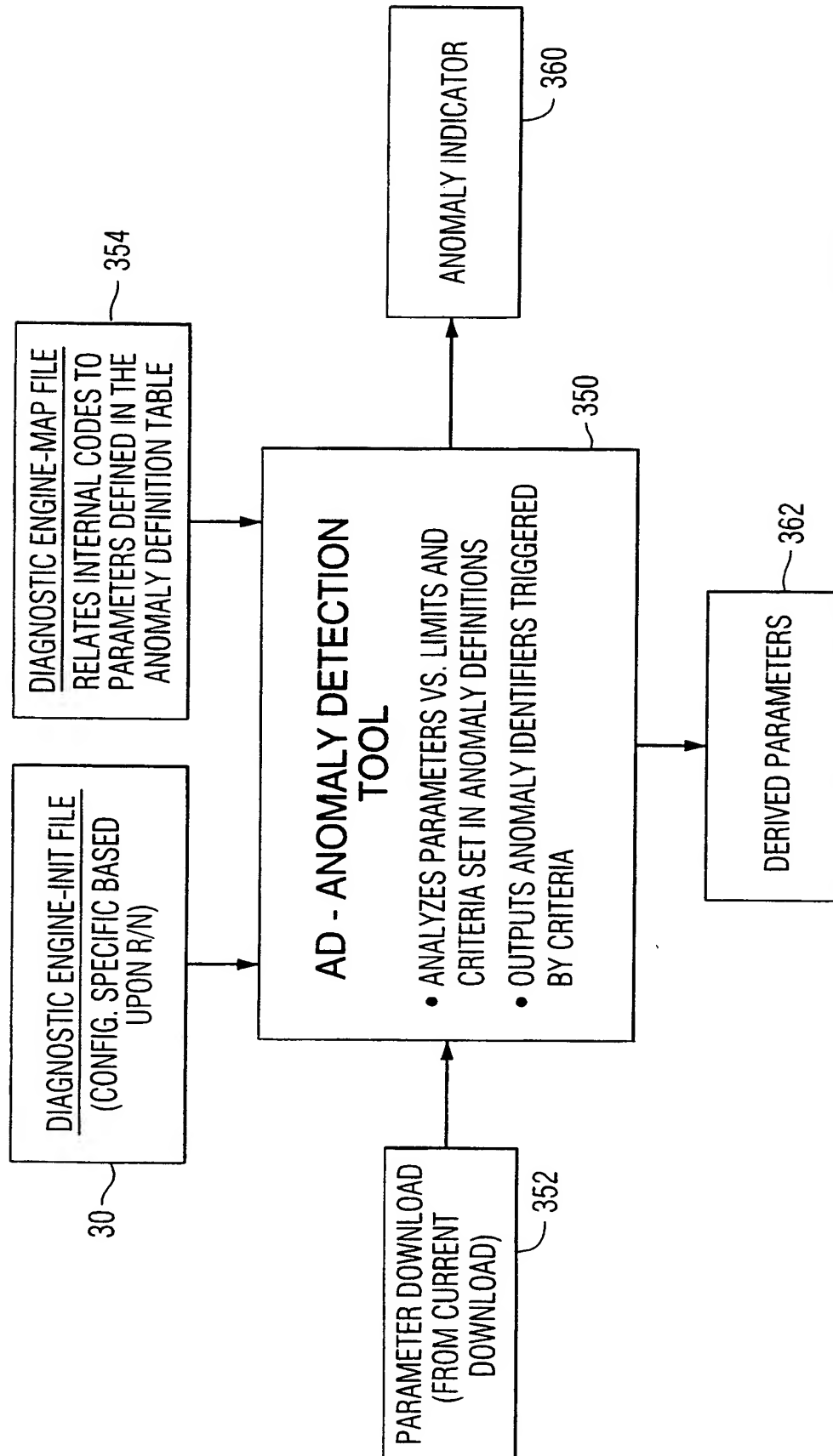


FIG. 14

17/19

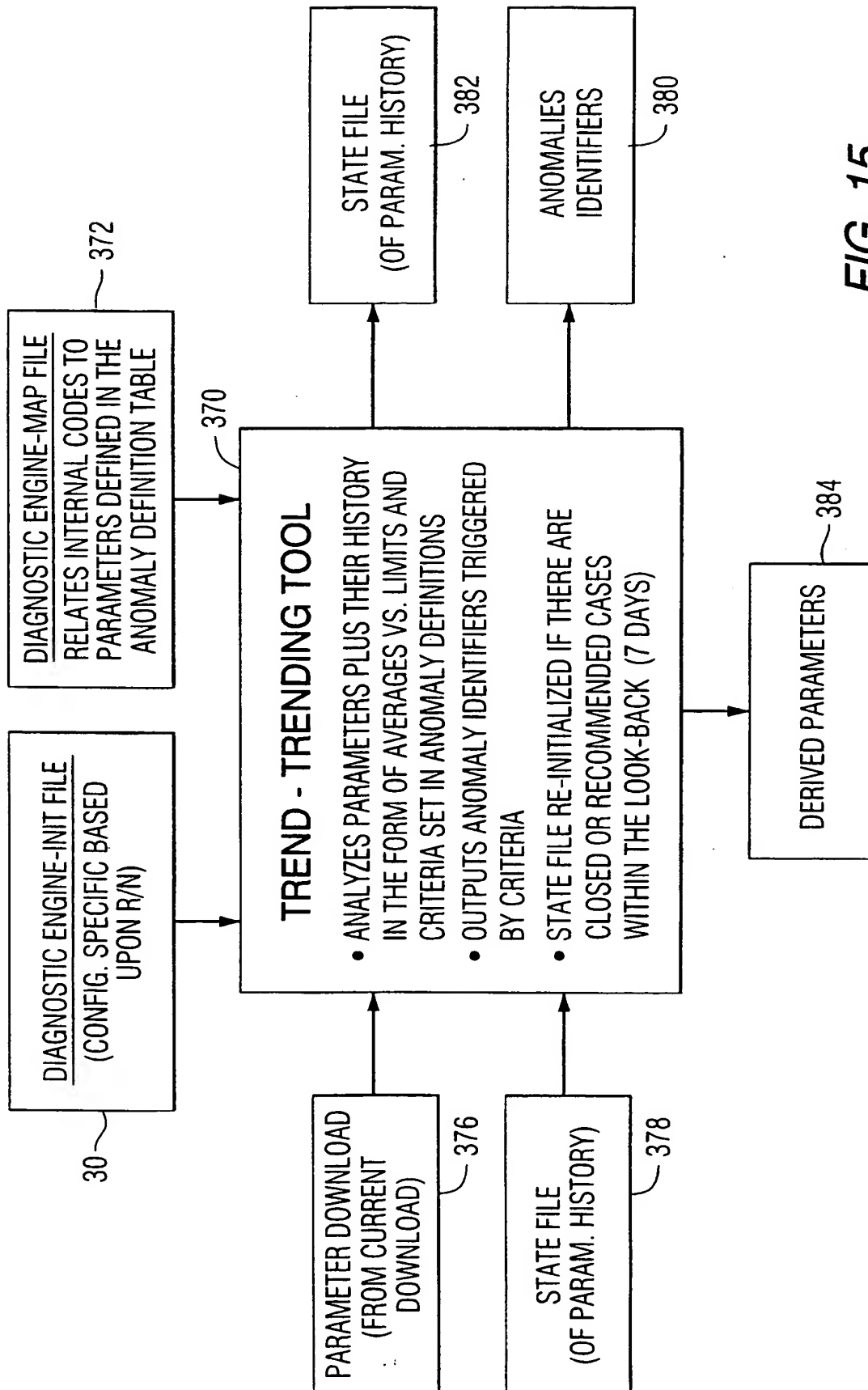


FIG. 15

18/19

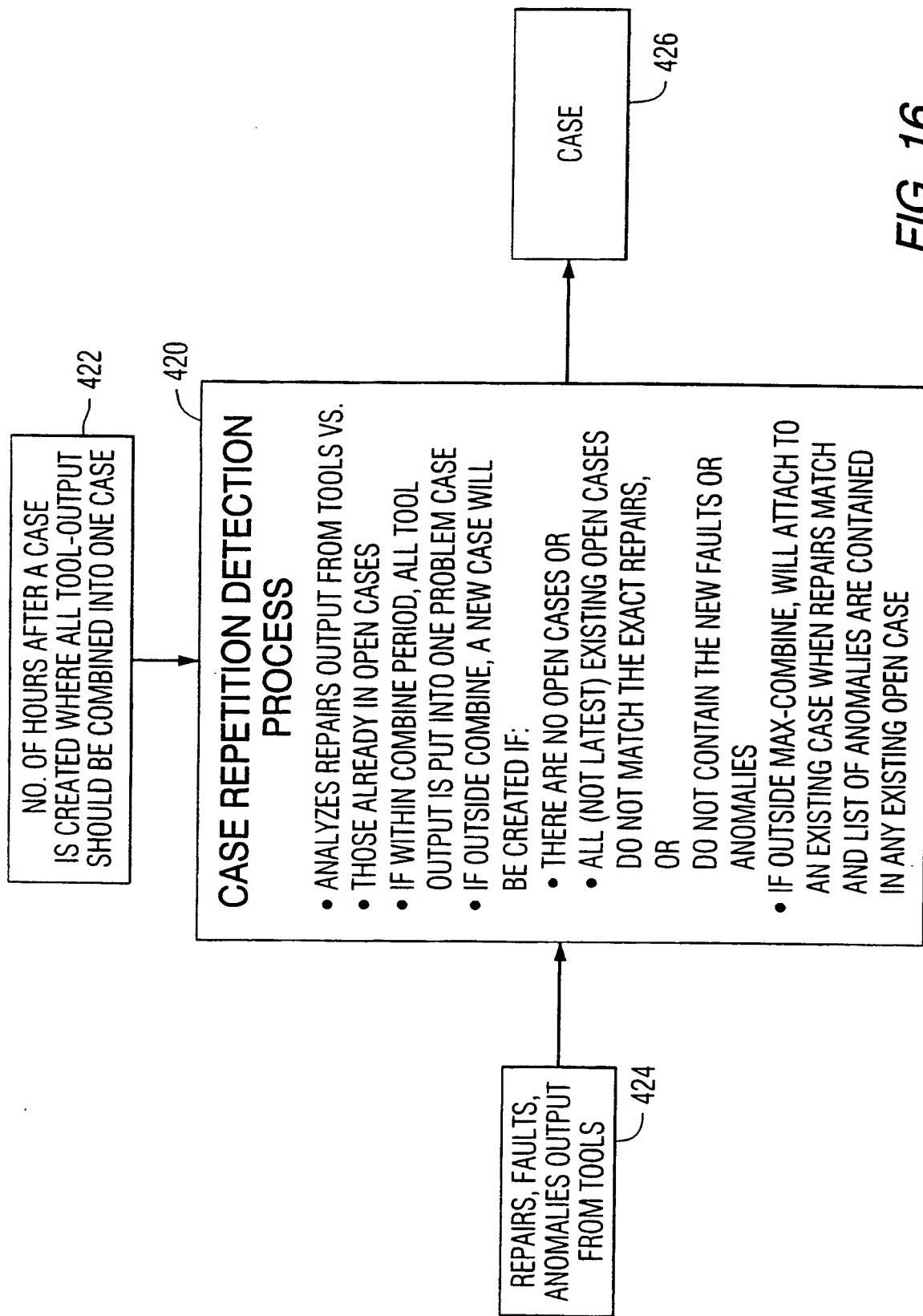


FIG. 16

19/19

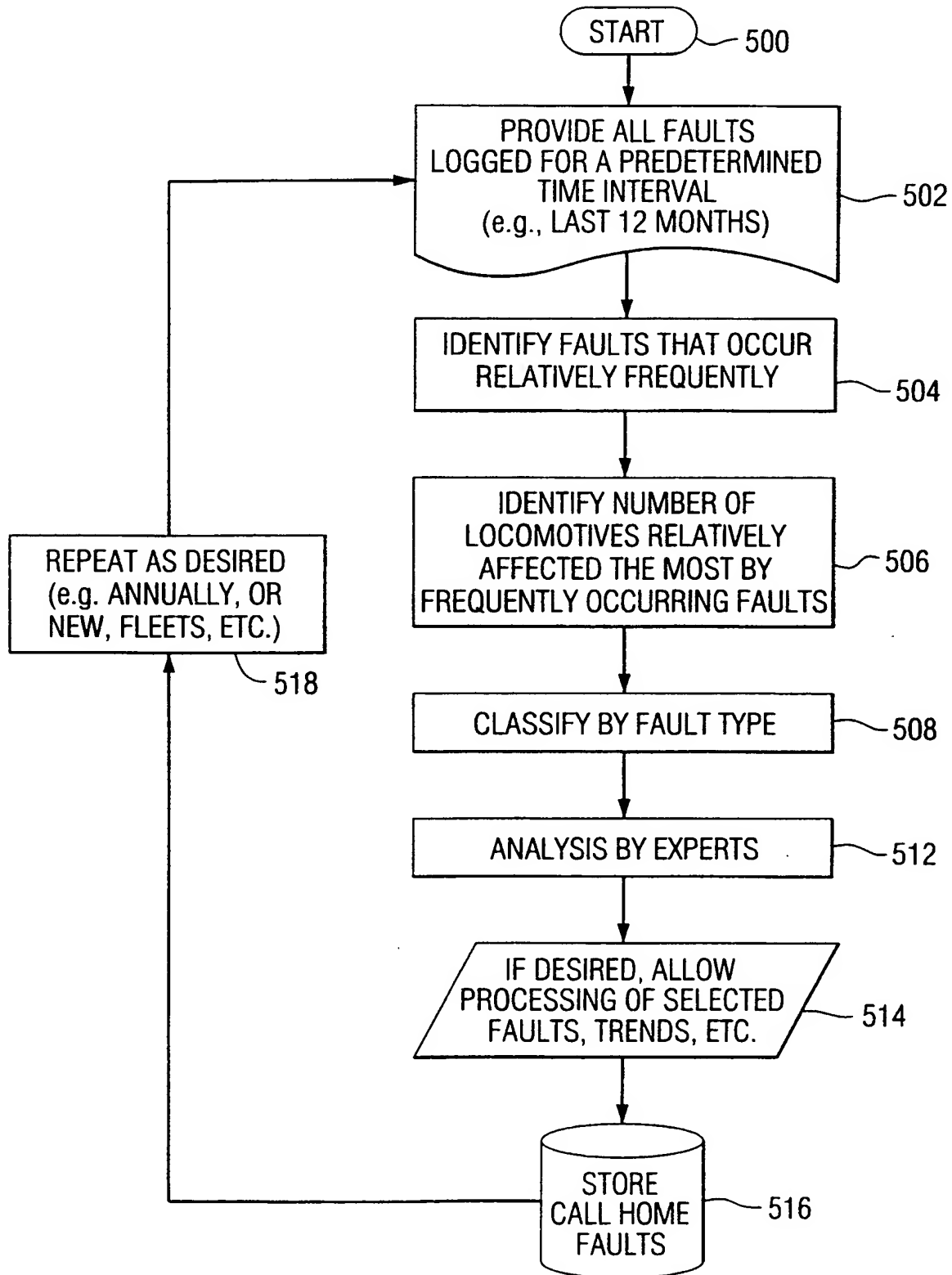


FIG. 17

INTERNATIONAL SEARCH REPORT

International Application No

PC:1/US 00/29439

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F11/25 G07C5/00 B61L27/00 G06F17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F G07C B61L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

PAJ, EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 97 13064 A (BERGSTROEM MAGNUS ;MOELLER PETER (SE); MUELLER JAN (SE); AALLEVING) 10 April 1997 (1997-04-10) abstract	1, 20, 24
A	PATENT ABSTRACTS OF JAPAN vol. 012, no. 053 (P-668), 18 February 1988 (1988-02-18) & JP 62 198943 A (NEC CORP), 2 September 1987 (1987-09-02) abstract	1, 15, 20
A	US 5 132 920 A (BELLOWS JAMES C ET AL) 21 July 1992 (1992-07-21) the whole document	1

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *G* document member of the same patent family

Date of the actual completion of the international search

9 February 2001

Date of mailing of the international search report

15/02/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Reekmans, M

INTERNATIONAL SEARCH REPORT

Int. Application No.

PCT/US 00/29439

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 737 215 A (SARANGAPANI JAGANNATHAN ET AL) 7 April 1998 (1998-04-07) the whole document ---	1
A	EP 0 810 558 A (HUGHES AIRCRAFT CO) 3 December 1997 (1997-12-03) the whole document ---	1
A	WO 90 09645 A (WARNERDALE LTD) 23 August 1990 (1990-08-23) the whole document -----	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/29439

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 9713064	A	10-04-1997	SE 510029 C EP 0853722 A JP 11513454 T SE 9503412 A US 6115653 A	12-04-1999 22-07-1998 16-11-1999 04-04-1997 05-09-2000
JP 62198943	A	02-09-1987	NONE	
US 5132920	A	21-07-1992	CA 1326555 A CN 1036647 A,B ES 2010403 A JP 2008999 A JP 2824524 B KR 9704261 B	25-01-1994 25-10-1989 01-11-1989 12-01-1990 11-11-1998 26-03-1997
US 5737215	A	07-04-1998	DE 19651986 A JP 9202218 A	19-06-1997 05-08-1997
EP 0810558	A	03-12-1997	US 5931877 A AU 689062 B AU 2221097 A CA 2205429 A JP 2977783 B JP 10143389 A	03-08-1999 19-03-1998 04-12-1997 30-11-1997 15-11-1999 29-05-1998
WO 9009645	A	23-08-1990	NONE	

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)